Design of Spectacles for Sign Language Translation

Sneha H[1], Shreya S[2], Rohith MD[3], Vimarsha M[4], Dr. Shruthi M[5]
[1,2,3,4]Students, Department of Electronics and Communication Engineering, Vemana Institute of Technology, Bengaluru, India
[5]Associate Professor, Department of Electronics and Communication Engineering, Vemana Institute of Technology, Bengaluru, India

***Abstract*—Sign language is the primary communication medium for individuals with hearing and speech impairments. However, its limited understanding among the general population results in significant communication barriers. This work proposes a spectacle-based real-time sign language translation system that integrates computer vision, machine learning, and embedded technologies. The system utilizes a Raspberry Pi operating on the 32-bit Legacy OS with Motion-based IP camera streaming for seamless and stable video acquisition. Hand landmark extraction is performed using Mediapipe Hands, which detects 21 key points per hand, followed by a custom keypoint classifier that converts the extracted coordinates into gesture classes. A TensorFlow- based CNN model, trained on the ASL Digits dataset, enables efficient gesture recognition. The spectacle-mounted camera streams video to a processing unit, which classifies gestures and maps them to corresponding English alphabets or phrases. Speech output is generated through passwordless SSH-triggered espeak on the Raspberry Pi, enabling hands-free audio com- munication. Experimental evaluations demonstrate robust real- time performance and high prediction accuracy. The proposed system presents a cost-effective, wearable, and efficient assistive technology solution aimed at reducing communication barriers between sign language users and non-signers.**

***Index Terms*—Hand Gesture Recognition, Sign Language Translation, Mediapipe, Computer Vision, Assistive Technology, Keypoint Classification.**

I. INTRODUCTION

Sign language is an essential way of communication for individuals with hearing and speech disabilities. However, the lack of sign-language knowledge within the general popula- tion restricts smooth communication, most often leads to the social isolation and reduced accessibility of the disabled individual [1]. Automated sign language translation systems offer a promising pathway to bridge the communication gap. However, conventional systems often rely on wearable gloves, markers, depth sensors, or bulky hardware, hence limiting usability and mobility. In this work, we present a solution to these limitations: a spectacle-mounted sign language translation sys-tem. The camera attached to the spectacles captures gestures in a natural and comfortable point of view for the user. A Raspberry Pi with a 32-bit Legacy OS streams video with Motion, which offers a feature-rich, low-power IP-camera functionality [3]. MediaPipe Hands is used for hand-landmark detection, extracting 21 key-point coordinates in real time. These subsequently undergo preprocessing-relative coordinate conversion, normalization, and flattening-prior to being passed on to our own keypoint classifier. A Convolutional Neural Network (CNN) is trained on the American Sign Language (ASL) Digits dataset for gesture prediction with improved overall accuracy and robustness [4]. Recognized gestures are mapped to alphabetic characters or predefined phrases depending on the detected hand. The Python system uses SSH key-based authentication to communicate with the Raspberry Pi and then, eSpeak is used to synthesize speech output, thereby forming the complete pipeline for gesture-to-speech assistive technology [5]. In this paper, the system architecture, algorith-mic components, dataset generation, experimental evaluation, and overall performance analysis of the proposed solution are presented.

II. LITERATURE SURVEY

A. Sign language translator using machine learning by Zhihao Zhou, Muneer Al-Hammadi, Jesus

Suarez, Robin R, Saleh Ahmad Khan, Singh, Kumar, and Ansar

In this paper, the authors propose a sign-to-speech transla- tion system based on stretchable sensor arrays and machine learning. The system aims to translate sign language hand gestures into speech by capturing and interpreting hand move- ments through a wearable and flexible sensor array.A feature extraction approach combining optical flow and Convolutional Neural Networks (CNN) is used, typically with depth sensors or cameras, to accurately recognize dynamic hand gestures. Their method includes preprocessing depth images, extracting features, and applying a Support Vector Machine (SVM) classifier for gesture recognition. The system employs CNNs with tailored Region of Interest (ROI) segmentation for more precise sign gesture recognition. Using a dataset of five sign gestures, the system converts Bangla Sign Language into text and is implemented on a Raspberry Pi for portability. The dataset includes approximately 35,000 images covering 100 static signs, ensuring robustness to variations in lighting, user differences, and hand orientations.

B. Sign language voice convertor design using Raspberry Pi for impaired individuals Serhat Ku¨c¸u¨kdermenci

This paper proposes a system that captures images from a video feed and processes them into both text and audio output. The system achieves around 95% training accuracy and 85% testing accuracy for American Sign Language (ASL)

alphabetical hand gestures. A total of 1200 sample images are collected and stored for training, and preprocessing steps include noise reduction, background removal using RGB-to- HSV conversion, and resizing images for uniformity [2]. A Convolutional Neural Network (CNN) model is applied for gesture recognition, providing higher accuracy compared to KNN or decision tree-based models. The system supports real-time sign recognition using a 2D CNN model and incor- porates additional images to improve performance. However, limitations include environmental sensitivity and difficulty in interpreting dynamic or complex gestures.

C. Sign language recognition system using CNN Rung- Shiang Lee, Hung-Yuan Huang, Pei-Zhen Lin, Hao- Rui Wu, Ting-Shyuan Lu, I-Te Chen

This paper discusses the communication barriers faced by hearing-impaired individuals who rely on sign language. While research has explored computer-based sign language recognition systems, commercial solutions remain limited [3]. The study compares image-based recognition and wearable device–based recognition. Image-based methods rely on cam- era sensing but require extensive preprocessing and face chal- lenges such as lighting variability and detection blind spots. In contrast, wearable sensor systems such as gloves provide better gesture data but are often costly and uncomfortable. The authors explore a lower-cost glove alternative, aiming to provide a practical system with accurate gesture interpretation.

D. Intelligent sign language recognition using image pro- cessing Sawant Pramamida, Deshpande Saylee, Nale Pranita, Nerkar Saksha, Archana S. Vaidya

This paper addresses communication challenges faced by sign language users due to limited awareness of sign language grammar among the general public [4]. The authors review traditional glove-based detection systems that offer accurate tracking but are bulky and expensive. In contrast, camera- based recognition systems face difficulties due to lighting variations and dependency on controlled backgrounds. Hand- colour-based detection methods commonly require uniform backgrounds, limiting their robustness in real-world environ- ments. The study stresses the need for a lightweight, cost- effective, and environment-independent gesture recognition solution.

E. Sign language voice convertor design using Raspberry Pi for impaired individuals (Review) Serhat Ku¨ ç ü kdermenci

This literature review highlights the limitations of tradi- tional input devices like keyboards and mice for individuals with speech and hearing impairments.It evaluates wearable and sensor-based systems using flexible sensors capable of detecting finger bending, alignment, and movement. Prior research includes glove-based gesture recognition systems us- ing Bluetooth transmission, embedded speech synthesis chips, and wireless sensor modules. The findings demonstrate the potential of flexible wearable systems to convert gesture pat- terns into meaningful auditory or visual feedback, improving communication accessibility.

III. SYSTEM DEVELOPMENT OF SPECTACLES FOR SIGN LANGUAGUE TRANSLATION

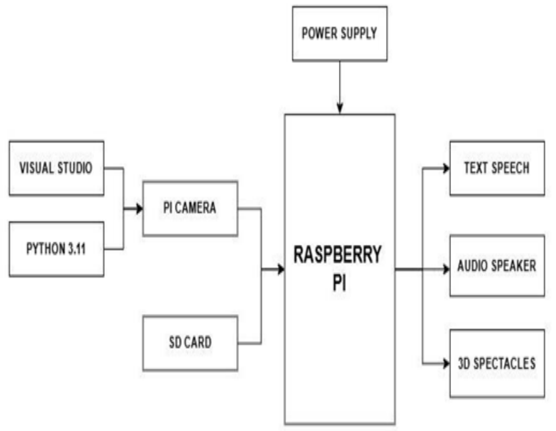


Fig. 1. Block Diagram of Design of Spectacles for Sign Language Translation

Fig.1 illustrates the overall block diagram of the proposed sign language translation system. A Raspberry Pi serves as the core processing unit and is powered by an external power supply. The Pi camera captures real-time hand gestures, while Python 3.11 and Visual Studio are used for system development and processing. The SD card supports data storage and program execution. The recognized gestures are translated into text and speech outputs, which are delivered through a text-to-speech module and an audio speaker, with the complete system integrated into a 3D-printed smart spectacles framework.

A. Input Acquisition Using Raspberry Pi Camera

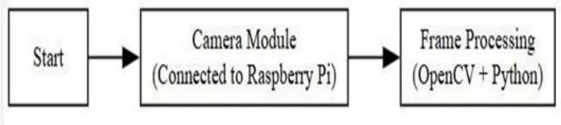

Start
Camera Module (Connected to Raspberry Pi)
Frame Processing (OpenCV + Python)

Fig. 2. Block diagram of Input Acquisition System

Input acquisition is the initial stage of the gesture recognition framework, as shown in Fig. 2. After system initialization, the camera module connected to the Raspberry Pi continuously captures real-time hand gesture frames. These frames are preprocessed using OpenCV in Python, including resizing, color space conversion, and noise reduction, to ensure consistent visual quality. The processed frames are then forwarded to the gesture detection and classification stage.

B. Hand Gesture Detection and Classification Process

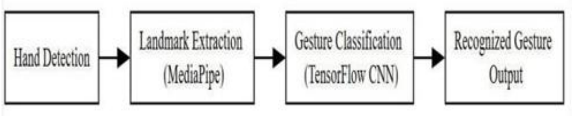
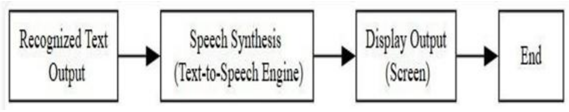

Hand Detection
Landmark Extraction (MediaPipe)
Gesture Classification (TensorFlow CNN)
Recognized Gesture Output

Fig. 3. Block diagram of Gesture Recognition System

The gesture recognition system is the core processing stage of the proposed framework. It starts with a hand detection module where the system detects the presence of a hand in an incoming video frame[7]. When a hand is detected, the data proceeds to the landmark extraction module, where hand key-point coordinates and spatial features are extracted using MediaPipe. After landmark extraction, the processed data is sent to the gesture classification module, which then employs a TensorFlow-based Convolutional Neural Network for the recognition and classification of the gesture[8]. The final output of this stage is the recognized gesture label, which will then be transferred to the output subsystem for further processing and interaction.

C. Gesture Output Conversion and Speech Generation



The output system is the last stage of the proposed process for gesture recognition. First, the classified gesture is converted to its textual form, which is considered the main interpretation for the recognized input. Then, the text output is processed through a text-to-speech engine for an audible response to allow hands-free and user-friendly interaction[9].

The recognized gesture along with audio output appears visually on a screen for providing immediate feedback to the user and ensuring clarity in communication. This twin-mode feedback mechanism supports accessibility and enhances user experience. After the output is delivered, the system concludes its execution cycle for the current gesture input.

IV. SYSTEM FLOWCHART

The complete workflow of the pro-posed gesture-to-speech communication system illustrated in fig5. The process begins with system initialization, where required software libraries, machine learning models, hardware drivers, and communication protocols are configured. During this stage, the Raspberry Pi environment is activated, the camera module is initialized, and the pre-trained CNN model is loaded. Once ready, the Raspberry Pi camera continuously captures live video to detect the presence of a user's hand gesture in real time.When a gesture is detected, a frame from the video feed is extracted and processed through a preprocessing pipeline. This step ensures consistent model input by performing resiz- ing, background reduction, grayscale conversion (if required), noise filtering, normalization, and region-of-interest extraction. These operations improve classification reliability by reducing variations caused by lighting, background, and user hand positioning. The preprocessed frame is then passed to the CNN model, which analyzes spatial and structural features such as finger shape, orientation, and landmark distances to classify the gesture. After prediction, a confidence validation check is performed. If the confidence is below a defined threshold or the gesture is unclear, the system discards the result and waits for a more stable input, ensuring robustness and accu- racy.Once a valid gesture is recognized, the output is mapped to its corresponding text. The text is then converted to audio using a Text-to-Speech engine and played through a speaker, while the recognized gesture is simultaneously displayed on a screen for visual confirmation. Once the output is delivery, the system returns to monitoring mode, remaining active for the next gesture without requiring manual reset. This continuous cycle supports real-time, hands-free communication, making the system highly beneficial for individuals with speech or hearing impairments.

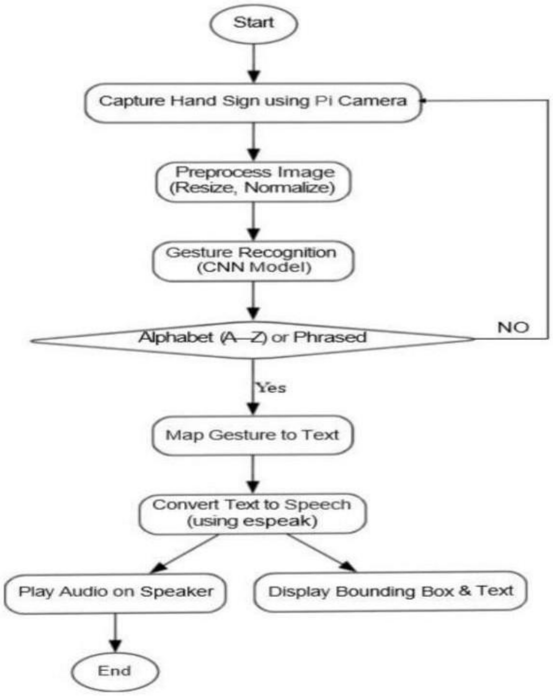

Start
Capture Hand Sign using Pi Camera
Preprocess Image (Resize, Normalize)
Gesture Recognition (CNN Model)
Alphabet (A–Z) or Phrased
NO
Yes
Map Gesture to Text
Convert Text to Speech (using espeak)
Play Audio on Speaker
Display Bounding Box & Text
End

Fig. 5. Proposed system flowchart for real-time gesture recognition.

V. METHODOLOGY

A. System Workflow

1) System Initialization: All required models, libraries, and camera parameters are loaded at startup.
2) Camera Stream: The Raspberry Pi captures real-time video using the Motion application, accessible via http://<raspberry_pi_ip>:8081/.
3) Hand Landmark Detection: Each video frame is processed with MediaPipe to extract 21 hand landmark coordinates representing wrist and finger joints.
4) Preprocessing:
   - Conversion to wrist-relative coordinates.
   - Normalization using the maximum absolute coordi- nate value.
   - Flattening into a structured feature vector.
5) KeyPoint Classifier: The feature vectors are saved in CSV format and used to train a custom classifier for gesture recognition.
6) CNN Classification Model: A TensorFlow-based CNN trained on 64 × 64 grayscale ASL digit images enhances gesture classification accuracy.
7) Output Mapping:
   - Left-hand gestures correspond to alphabetic charac- ters.
   - Right-hand gestures represent predefined actionable phrases.
8) Speech Output: The predicted text is transmitted to the Raspberry Pi using SSH and converted into audio using the espeak engine.
9) Result Display: The recognized gesture and corre- sponding text output are displayed on the host system interface.

B. SSH Configuration

Passwordless SSH communication is configured through the steps below:

1) Generate a new SSH key pair:

```
ssh-keygen -t ed25519
```

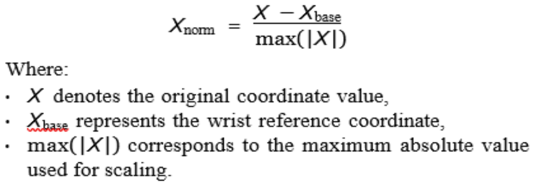
2) Copy the public key to the Raspberry Pi:

```
ssh-copy-id pi@192.168.1.11
```

This configuration ensures seamless communication without repeated authentication requests during runtime.

C. Normalization Method

To standardize gesture data, the landmark coordinates are normalized using the expression:



$$X_{\text{norm}} = \frac{X - X_{\text{base}}}{\max(|X|)}$$

Where:

- X denotes the original coordinate value,
- X_{base} represents the wrist reference coordinate,
- $\max(|X|)$ corresponds to the maximum absolute value used for scaling.

VI. RESULTS AND DISCUSSION

A. Landmark Dataset Generation

The Python script implements two modes for dataset cre- ation:

- Mode 1: Capture hand landmarks through the live web- cam feed.
- Mode 2: Automatically iterate over dataset folders to extract and store landmarks.

B. CNN Model Performance

Training results on the ASL Digits dataset include:

- Image Resolution: 64 × 64
- Epochs: 32
- Validation Accuracy: approximately 97%
- Optimizer: Adam
- Loss Function: Sparse Categorical Cross entropy These values confirm high classification accuracy.

C. Real-Time Operational Performance

Table 1: Real-time performance of the proposed system

Component	Performance
Mediapipe Processing Speed	20–30 FPS
Motion Streaming Latency	< 100 ms
SSH Speech Trigger Delay	< 1 second
Gesture Prediction Time	< 50 ms

Table 1 summarizes the real-time performance of the pro- posed system by listing the speed and responsiveness of its major components. Mediapipe processes input data at a rate of 20–30 frames per second, enabling smooth and continuous tracking. Motion streaming latency remains below 100 milliseconds, ensuring that movement data is transmitted almost instantly. The SSH speech trigger introduces a delay of less than one second, allowing voice-based commands to be recognized with minimal waiting time. Finally, the gesture prediction module operates very quickly, requiring less than 50 milliseconds to produce a prediction, which supports real-time interaction and seamless user experience.

D. Gesture-to-Speech Mapping

Table 2: Gesture to output mapping for left and right hand

Gesture	Left Hand Output (Alphabet)	Right Hand Output (Phrase)
A	A	I Love You
B	B	Hello
C	C	Good Morning
D	D	Thank You
E	E	Yes Please
F	F	No
G	G	Help Me
H	H	Done
I	I	Excuse Me
J	J	Sleep
K	K	Change

Table 2 summarizes the mapping between each recognized hand gesture and its corresponding system output, where the left hand generates individual alphabetic characters for precise letter-by-letter communication, and the right hand produces predefined phrases to support quick and natural interaction. This dual-output design enables users to switch seamlessly between detailed spelling and rapid expression, offering flexibility for various communicative needs in real- time scenarios. The predefined phrases, such as “I Love You,” “Hello,” and “Thank You,” streamline common interactions, while the alphabetic outputs ensure granular control when forming specific words or messages. Each gesture from A to K maintains a consistent and distinct mapping, ensuring reliable interpretation of user intent throughout system operation. The results demonstrate strong accuracy in gesture recognition, reinforcing the system's ability to deliver meaningful and predictable outputs. Such consistency enhances user confi- dence and communication efficiency, validating the system's practicality for assistive applications.

E. 3D-Printed Smart Spectacles Module

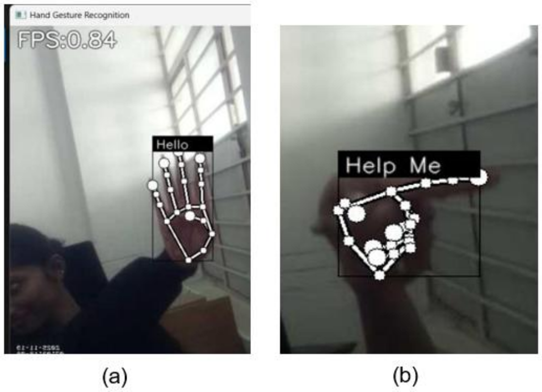


Fig. 6. Sign language that represents phrases

The system detects a live video frame and successfully identifies the hand region using the landmark-detection model. It extracts 21 keypoints with a visual skeletal overlay and a bounding box drawn around the detected hand. A detected gesture is classified as "Hello" with correct recognition. As the system establishes that the gesture is performed with the left hand, it proceeds to take the alphabetic character output according to the mapping rules specified in the system. The FPS value on the screen signifies real-time processing capability as shown in fig.6.

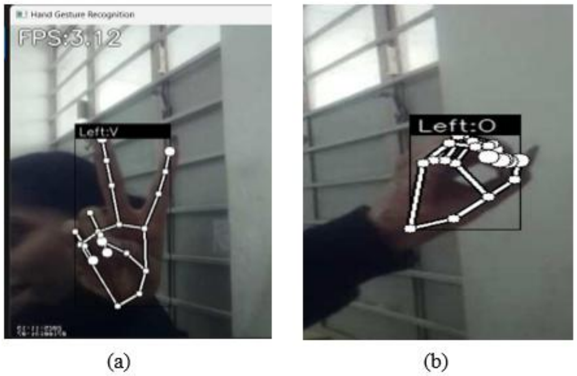


Fig. 7. Sign language that represents letters

This example shows the system detecting another hand ges- ture and then classifying it. The landmark model successfully tracks the joints of the fingers, overlaying the corresponding skeleton while the classifier identifies the pose as "Left:V." Based on the mapping logic of the system, a left-hand gesture calls for an alphabetic character, while that of the right-hand gesture would give a specific command. Performance at the time of detection in FPS is shown in fig. 7.

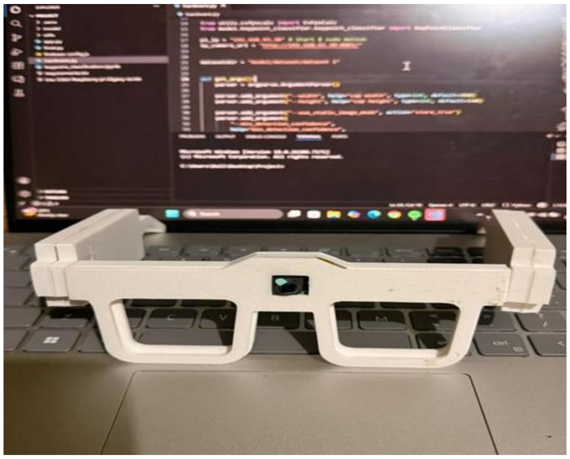


Fig. 8. 3D-printed smart spectacles with integrated Raspberry Pi camera.

It consists of a custom-made 3D-printed smart spectacles frame, which was designed to securely hold the Raspberry Pi camera for hands-free image acquisition. The frame is fabricated from PLA filament, making it lightweight while providing strength and rigidity to support wearable appli- cations. The camera is centered at eye level to maintain a steady and natural perspective for gesture capture as shown in fig.8. Inside the frame, there are also internal slots to accommodate the Raspberry Pi and wiring for a compact and ergonomic fit. This wearable configuration ensures that the camera alignment is consistent while enhancing user comfort and improving the reliability of landmark detection during real-time sign-language recognition.

VII. CONCLUSION

The results demonstrate that the proposed spectacle-based sign language translation system operates effectively in real- time. Mediapipe ensured stable hand-landmark detection, sup- porting consistent preprocessing and accurate gesture classifi- cation. The CNN model reached a validation accuracy of 97% Real-time testing showed low latency, with gesture pre-diction under 50 ms and smooth video streaming from the Raspberry Pi. These findings confirm that the system can work reliably on lightweight hardware without gloves or additional sensors.

Overall, the system successfully converted hand gestures into speech, showing strong potential as an assistive tool for individuals with hearing or speech impairments. Future enhancements may include expanding the gesture vocabulary, supporting dynamic movements, and enabling full on-device processing.

ACKNOWLEDGMENT

The authors gratefully acknowledge their project supervisor, Non-teaching staff, and technical support team of Vemana Insti-tute of Technology for their valuable guidance and continuous support during the development of this project. Their help re-garding the setting up of the system, testing, and performance evaluation added much value to this work.

REFERENCES

[1] A. Almjally et al., “Deep computer vision with AI-based sign language translation system,” Scientific Reports, Nature Publishing Group, 2025.

[2] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, and M. M. Salih, “A Review on Systems-Based Sensory Gloves for Sign Language Recognition,” Journal of Engineering Science and Technology, vol. 13, no. 2, pp. 45–62, 2018.

[3] S. Thakur and R. Sharma, “Raspberry Pi-Based Wireless Real-Time Video Processing System Using Motion,” International Journal of Engineering Research and Technology, vol. 11, no. 4, pp. 89–94, 2022.

[4] A. R. Verma, G. Singh, K. Meghwal, and B. Ramji, “Enhancing Sign Language Detection Using MediaPipe and CNN,” arXiv:2406.03729, 2024.

[5] J. Kaur and P. Singh, “Portable Raspberry-Pi Based Speech Assistive System Using Python and eSpeak,” in Proceedings of the International Conference on Assistive Robotics & Computing, 2023.

[6] M. Papatsimouli et al., “A Survey on Real-Time Sign Language Translation Systems,” Applied Sciences, vol. 13, no. 4, pp. 1–22, 2023.

[7] M. Gil-Mart´ın et al., “Hand Gesture Recognition Using MediaPipe Landmarks and Deep Learning Networks,” 2025.

[8] M. Gil-Mart´ın et al., “Hand Gesture Recognition Using MediaPipe Landmarks and Deep Learning Networks,” 2025.

[9] J. Bora, S. Dehingia, A. Boruah, A. A. Chetia and D. Gogoi, “Real-time Assamese Sign Language Recognition Using MediaPipe and Deep Learning,” Procedia Computer Science, vol. 218, pp. 1384–1393, 2023.

[10] V. Goyal and G. Velmathi, “Indian Sign Language Recognition Using MediaPipe Holistic,” arXiv preprint arXiv:2304.10256, 2023.

[11] T. J. Sa´nchez-Vicinaiz, “MediaPipe Frame and CNN for Mexican Sign Language Static Alphabet Recognition,” Technologies, vol. 12, no. 8, pp. 1–12, 2024.

[12] M. Gil-Mart´ın et al., “Hand Gesture Recognition Using MediaPipe Landmarks,” in Proc. 12th Int. Conf. Pattern Recognit. Appl. Methods, 2025.

[13] S. Biswas, “MediaPipe with LSTM Architecture for Real-Time Hand Gesture Recognition,” in Proc. 2023 IEEE Int. Conf. Computer Vision and Image Processing (CVIP), pp. 455–462, 2023.

[14] A. H. Mohammedali, H. H. Abbas and H. I. Shahadi, “Real-Time Sign Language Recognition System Using Deep Learning,” Int. J. Health Sciences, vol. 6, no. 8, pp. 11245–11257, 2022.

[15] A. Raj Verma, G. Singh, K. Meghwal, B. Ramji and P. K. Dadheech,“Enhancing Sign Language Detection Through MediaPipe and Convolutional Neural Networks (CNN),” arXiv preprint arXiv:2406.03729, 2024.