A Real-Time Sign Language Recognition and Speech Conversion System on Smart Wearable Devices

Anila Kuriakose, Leuwin M S, Harry Joby, Mohammed Naeem P A, Don S Sebastian
Department of Electronics and Communication Engineering
Rajagiri School of Engineering and Technology
(Autonomous)
Ernakulam, India

Abstract—Daily interactions between hearing-impaired individuals and those unfamiliar with sign language often face communication hurdles. To address this, we propose a lightweight, real-time system that translates sign language into audible speech using computer vision technologies. At the heart of the system is a Convolutional Neural Network (CNN), trained using the Keras training model, that classifies hand gestures into corresponding sign language symbols. To enhance accuracy and responsiveness, the system is integrated with OpenCV and MediaPipe for real-time hand tracking and gesture preprocessing. Once a gesture is recognized, it is converted into speech through a text-to-speech (TTS) system, enabling users to communicate via audio in real time. The entire system is implemented on a compact hardware platform that includes a Raspberry Pi 5, a small camera module, and Bluetooth-enabled audio output—housed in wearable smart glasses. This setup ensures portability, efficiency, and seamless communication, offering a meaningful step forward in assistive technology for the hearing-impaired community.

Index Terms—Sign Language Recognition, Real-Time Processing, Computer Vision, MediaPipe, OpenCV, Text-to-Speech, Assistive Technology.

I. INTRODUCTION

Communication is a critical component in the human life perspective, but some people have problems expressing their ideas in daily conversations. Deaf or mute people do not have the ability to hear and/or speak and face great issues in communicating with other people. People with disabilities find it very difficult to articulate their thoughts and ideas perfectly. To bridge this gap, researchers have explored a variety of technological solutions—ranging from sensor-equipped gloves to advanced image processing and wearable assistive devices[1][2]. Among these innovations, vision-based Sign Language Recognition (SLR) systems have shown great promise due to their non-intrusive design and ease of real-world application[3].

Modern advances in deep learning and computer vision (with libraries like OpenCV, MediaPipe , and TensorFlow) have introduced a variety of methods for classifying hand gestures in real time. Most of the works in this field adopt Convolutional Neural Networks (CNNs) to capture representative features from images and to create a stack of collected gesture words to model the temporal relation between gestures[4][5]. More sophisticated methods with sign language concepts through context-aware interpretations like transformer-based encoder-decoders are also studied in the literature[6],[7].

In this paper, we propose an accurate, lightweight, and real-time SLR system designed using OpenCV and MediaPipe for precise hand tracking and a custom-trained CNN to identify the hand gestures. These identified words are stacked together to form a sentence, and then when the stop sign is shown, it acts as a full stop, and the sentence is framed word by word. Once recognized, the stacked words are converted into natural-sounding speech using a neural text-to-speech (TTS) engine with voice cloning capabilities, which can be further expanded based on users needs. Prioritizing low-latency processing and minimal hardware requirements, the system avoids heavy Transformer-based pipelines[7],[8]. That is basically making the system light enough to run on any system. Instead, it runs efficiently on edge devices—specifically, a Raspberry Pi 5 embedded in a prototype of smart glasses. The glasses include a built-in camera module and Bluetooth audio, enabling real-time, hands-free communication. This brings more convenient and portable sign recognition technology, taking us one step closer to a more inclusive society for people with hearing and speech impairments, making their voices heard in life as well as sharing their thoughts and beliefs with others.

II. RELATED WORKS

Sign Language Recognition (SLR) has been a major frontier in recent research development, combining human-computer interaction, computer vision, and deep learning. The problem of converting sign language into words or speech has received attention using different approaches. Previous designs essentially adopted systems of wearable sensors, which are gesture-based systems with integrated inertial measurement units (IMUs), such as an accelerometer, gyroscope, and flex sensor, for the collection of hand movements[9]. The downside of these systems is that they are expensive to operate, invasive, and not practical for real-time use.

Vision-based solutions have played a critical role in addressing communication challenges for the hearing and speech impaired by applying machine learning techniques like Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN), and Hidden Markov Models (HMMs) to classify hand gestures[10]. While these early methods offered foundational insights, they often fell short in real-time performance and struggled to adapt to complex, fast-changing environments.

Deep learning has fundamentally era-defined the field as its popularity has grown thoroughly. CNNs have been the popular choice for extracting spatial features from image-based sign gestures, providing increased precision and robustness[6]. To go beyond that, hybrid models—admixtures of CNNs and LSTM networks—have improved recognition performance by considering temporal flow as well as sign context in dynamic sign sequences[11]. These advances represent significant progress in developing responsive and expandable

Heavily using benchmark datasets like RWTH-PHOENIX-Weather, the Indian sign language dataset, and ASL has gradually trained deep learning models [5]. Real-time systems have integrated OpenCV and MediaPipe for hand tracking and keypoint detection, enabling efficient feature extraction[13],[14]. Advanced architectures, including 3D CNNs and attention-based Transformer networks, have been explored in frameworks like Sign2Text and DeepHand, significantly improving temporal modeling and comprehension of sequential signs[6],[8]. However, many existing systems primarily focus on isolated word or letter recognition and struggle with natural sentence formation, which is crucial for effective communication.

Furthermore, because of latency problems, power limitations, and computational limitations, deploying SLR systems on embedded and wearable platforms continues to be difficult. The use of Raspberry Pi,

Arduino, and Edge TPU accelerators for gesture recognition has been the subject of numerous studies; however, the majority of these are restricted to offline processing or single-word classification, with little attention paid to real-time deployment and user-centric design[3],[4]. Some prototypes feature head-mounted displays or glove-based sensors, but these frequently detract from portability or usability[1][2]. Additionally, there aren't many systems that integrate real-time speech synthesis and gesture recognition in a small, wearable package. This gap drives our end-to-end development of a smart glasses prototype that uses a Raspberry Pi 5 and Bluetooth audio output to translate speech and detect signs in real time.

Building on recent breakthroughs in deep learning and computer vision, our approach is to introduce a real-time sign language recognition system that combines both accuracy & efficiency. Using OpenCV and CVZone for precise hand tracking, the system features a two-stage deep learning pipeline. First, a CNN model[7] identifies individual letters from hand gestures. The system usability is also maintained in this system. The letter sequences are subsequently fed to a Transformer-based language model that can reason over full words and sentences, which achieves a more natural, sentence-level comprehension of the text[8].

To enhance the user experience, we use a neural text-to-speech (TTS) engine with voice cloning to generate optimal quality, customized audio output[15]. Made with smart wearable devices in mind, the service needs nothing more than a camera and speaker to work—no clunky hardware involved. This low-latency and portable system enables continuous sign language recognition and real-time translation and can be used as a supplement to assist such spread-out people, which can provide more convenience for people who are deaf and/or mute.

III. PROPOSED SYSTEM

The proposed system offers a highly efficient real-time sign language recognition (SLR) solution powered by a simple camera setup. It operates through a six-stage processing pipeline designed to balance speed, accuracy, and usability. The system was tested in two settings: embedded deployment on a desktop-based development and a smart wearable prototype.

During the software development, the cvzone HandDetector module and MediaPipe's hand-tracking framework were initially utilized to train and validate gesture recognition on a 1920x1080 pixel webcam. One hand at a time was detected by the system, which had a detection confidence threshold of 0.8.A 720p USB camera built into the frame of the smart glasses and linked to a Raspberry Pi 5 was used in place of the original camera for real-world deployment. From the user's point of view, the small camera was positioned for the best hand tracking. The system maintained steady hand landmark detection that was appropriate for live prediction and sentence construction even under changing motion and lighting conditions.

To reduce computational load, the system focuses on a single dominant hand. Once detected, it extracts an adaptive region of interest (ROI), which is resized to 300×300 pixels and enhanced through brightness normalization, Gaussian blur, and histogram equalization for optimal clarity.The refined image is then passed into a TensorFlow-trained Convolutional Neural Network (CNN), which classifies the gesture and assigns a confidence score. To minimize errors caused by sudden hand movements, the system incorporates a deque-based buffer. This applies temporal smoothing using majority voting and confidence-weighted aggregation, significantly reducing false positives.

Fig. 3.1 : System Block Diagram

A. System Design and Implementation

The system is designed to achieve live and reliable sign language recognition with a vision-based approach with a regular camera that is expandable to higher resolution if desired. The interconnected modules cooperate and operate to identify hand gestures, translate them into intelligible text, and translate it into hand shape using the speech synthesis handler.

At the core of the system is the hand detection module, which identifies the presence and position of a hand in every video frame. This is accomplished using OpenCV and CVZone's HandDetector, which leverages an optimized implementation of MediaPipe's hand-tracking pipeline. By extracting key hand landmarks and providing accurate bounding box coordinates, the system maintains stable and consistent detection—even under varying lighting or cluttered backgrounds.

Each hand was cropped and then resized/grayscaled to a 300x300 pixel frame size and normalized with the objective of getting better consistency in frame size and lighting. The noise in the images was suppressed by applying a small Gaussian blur to the image prior to feeding the image to the classifier. All that preprocessing was necessary so the classifier could perform well and fast no matter the lighting. Once a hand is detected, the gesture classification module processes the extracted hand region using a deep learning model built with TensorFlow. Each gesture is classified into a predefined word, and low-confidence predictions are automatically filtered out to boost reliability and reduce false positives.

To further enhance accuracy, the system integrates a temporal smoothing mechanism. A deque-based sliding window keeps track of recent gesture predictions, and a weighted majority voting algorithm selects the most consistent and confident output. This method helps eliminate errors caused by brief hand motion glitches or occlusions, ensuring smoother and more stable recognition. Once the gestures are recognized, they are appended to an already existing sentence using the text conversion module. The system discards repetitive terms and preserves good surface form, which facilitates the coherent and natural-sounding sentence generation as the dialogue develops. Upon detection of a predetermined "Stop" gesture, the system activates the text-to-speech (TTS) module. The final sentence is converted to spoken language by the neural TTS engine , pyttsx3 for a smooth voice output in HRI applications such as real-time situation recognition.

B. Hardware and Software

The system is developed by using common hardware and software building blocks. The hardware includes a Raspberry pi 5 board, a standard webcam or USB camera, and a Bluetooth speaker for audio output. The software component uses Python as the base programming language, coupled with the basic libraries like OpenCV, cvzone, TensorFlow, and pyttsx3 for live processing.

Our sign language recognition system is built on a robust software stack designed for efficient real-time processing, gesture recognition, and speech synthesis. The software that is mainly used in this system is important Python libraries like OpenCV, cvzone, TensorFlow Lite, Keras, and pyttsx3. For the process

of hand detection and preprocessing, OpenCV and cvzone can be used. TensorFlow Lite plays a crucial role in the deep learning model interface on edge devices like the Raspberry Pi 5. The lite model of TensorFlow is being used here so that the system is a lightweight, real-time system and can be run on any system with ease. The Keras model is the model that is obtained after training the custom dataset, and using this model, we classify the gestures in the real-time run. Threading is used to ensure smooth execution of parallel tasks, that is, multiple processes can be handled by this system at once. Detected words were transformed using the inbuilt pyttsx3, which converts the detected gestures into audio, which is then let out through the Bluetooth headphone as output.

The proposed model is implemented on a Raspberry Pi 5 hardware with a quad-core Cortex-A76 processor and 8GB of RAM, and the running OS is a Linux-based one which is suitable to handle the computer vision tasks and the deep learning inference. HD 720p camera is used for accurate gesture tracking and Bluetooth TWS earbuds are used for audio output. This system combines a wearable smart glasses-style design with real-time hand tracking, dynamic sentence generation, and natural speech synthesis, ensuring a very natural and effective means of communication between people.

As an input device, we use a 720p USB camera, which is mounted on the smart glasses frame to capture live hand gesture input. Its compact size and wide-angle field of view make it suitable for wearable integration. As an output device, we used Bluetooth TWS speakers, which are connected wirelessly to the Raspberry Pi to deliver the synthesized speech output in real time, providing a hands-free communication experience. Use of TensorFlow Lite for model fine-tuning enabled low-latency gesture recognition and very clear real-time speech synthesis. The Raspberry Pi was powered using a portable power bank, enabling complete mobility and untethered operation of the smart glasses.

Fig. 3.2. : Hardware implementation

IV. DATASET AND TRAINING DETAILS

The training set for the sign language recognition model was created using a webcam using OpenCV. A sequence of hand gesture images corresponding to each word was captured with a custom Python script, which used the cvzone.HandTrackingModule module from which the handdetector function was

imported for hand detection and isolation. The images were resized to 300x300 pixels and saved into corresponding folders based on the pre-defined words. For example, to get a word output like "morning" the script would record hand images as the user shows the related sign and save them to a given directory. This process was repeated for many words including ‘bad’, ‘good’, ‘hello’, ‘I love you’, ‘morning’, ‘no’, ‘sorry’, ‘stop’, ‘thank you’, ‘where’, ‘why’ and ‘yes’.There were more than 60 samples for each word to have a sturdy model training process.

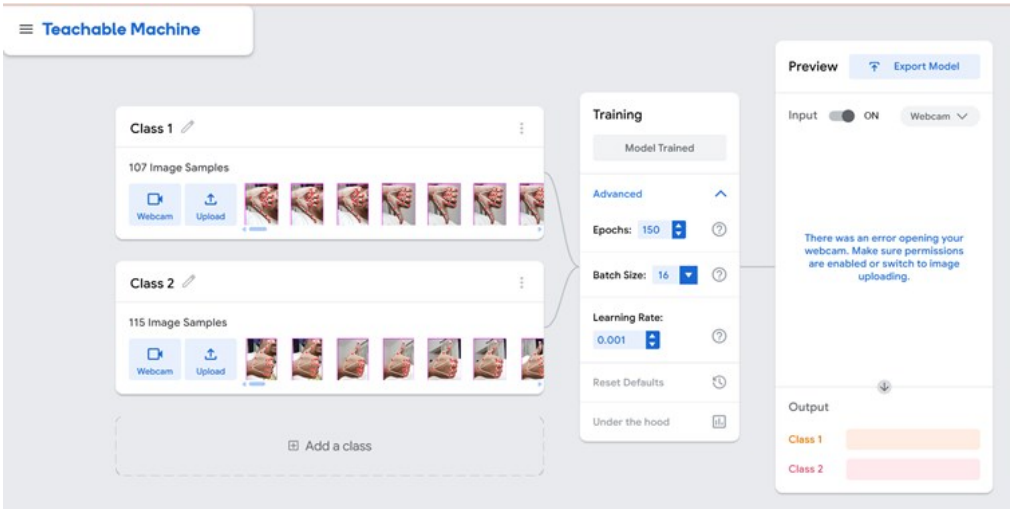

Fig. 4.1. : Training Process

The collected dataset was trained with Google Teachable Machine, which is an online training service provided by Google. A deep learning model using Convolutional Neural Networks (CNNs) was trained using this service. The generated model was exported as Keras and integrated to the project.

V. RESULTS & ANALYSIS

The developed sign language recognition system was installed successfully on a Raspberry Pi 5 embedded platform with an external camera (for video input) and headphones (through which the system outputs spoken language in real-time). The system recognizes hand gestures, automatically categorizes them into predefined sign language words, and translates them into text and speech in real time. The performance of the models was assessed in terms of accuracy, response time and user experience.

The hand detection and recognition system successfully detected hand gestures in real-time in a controlled environment. By using the TensorFlow Lite model, we successfully classified hand gestures with an average score of 96%. The smoothing algorithm was adjusted for high order misclassifications such as sudden hand movements and/or brief occlusions. The system was evaluated in various lighting and background clutter situations. For consistent lighting, the system performed well with stable recognition, whereas some inaccuracies were observed in dim or uneven lighting due to poor hand landmark detection. Background clutter and minor misclassifications were observed when the hand was not clearly distinguishable from the background.

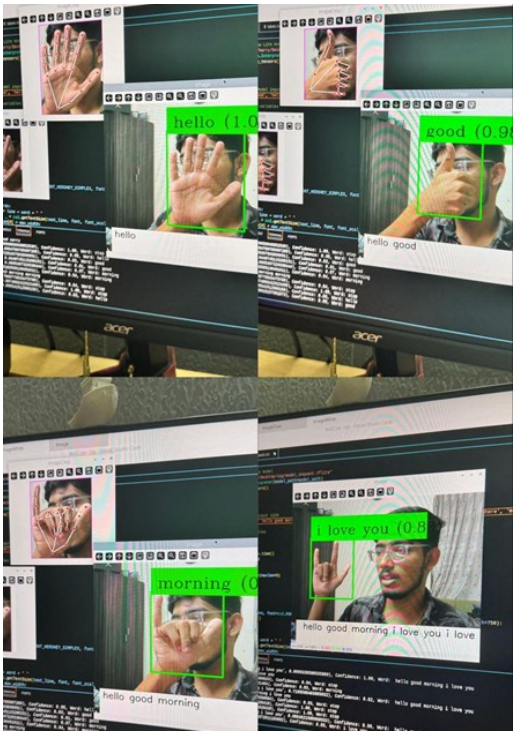

Figure 5.1: Sentence formation

When an analog gesture was identified, the corresponding word was shown on the computer monitor and became part of a sentence. The system did successfully build meaningful sentences with increasing data. Upon detecting the ‘stop’ gesture the sentence was uttered using headphones. The text-to-speech output was clearly audible and intelligible. There were minimal latencies observed eliciting a speech-gesture response body and for speech produced as a result of perceiving speech gesture.

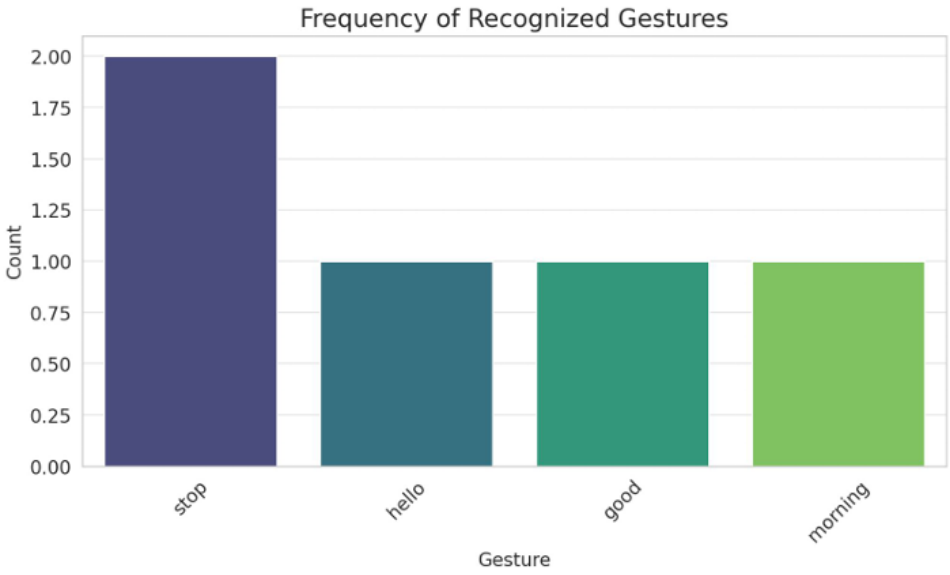

Fig. 5.2.: Frequency of recognized gestures

To understand which gestures were detected the most, the frequency of the recognized gestures was plotted in Fig 5.2. Uneven distribution can indicate model bias, meaning some gestures might be more easily recognized than others. The confidence score distribution of the proposed system is given in Fig. 5.3 which shows the degree of confidence in recognizing each gesture. All gestures have confidence scores above 0.965 which demonstrates strong classification performance. High median value shows the system is stable and reliable.The relatively low value of the 'morning' gesture shows it might be confused with other gestures and need better training data.

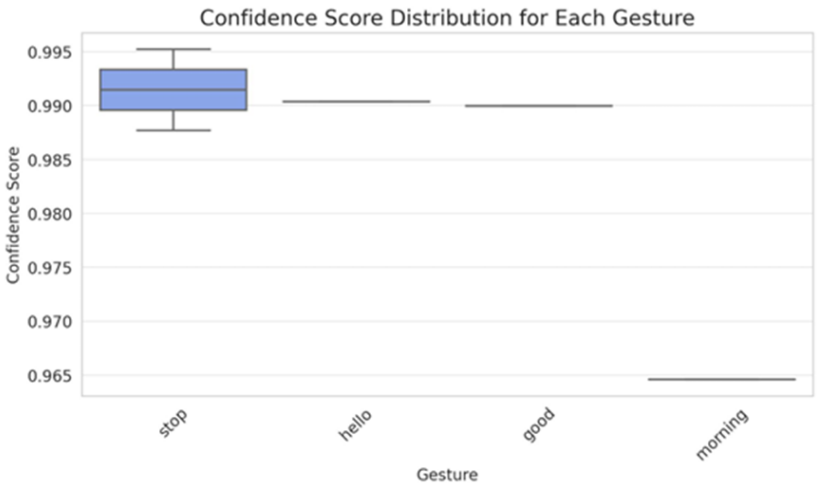

Figure 5.3 : Confidence score distribution for each gesture

To check whether the model processes the gestures efficiently and to check its system performance, the variation of processing time was plotted for various timestamps. The model exhibits relatively stable and efficient performance. The processing times varied between 0.102 s and 0.122 s, which shows that the system is capable of handling real-time gesture recognition.The consistent low latency suggests good optimization and suitability for real time applications like sign language interpretation.

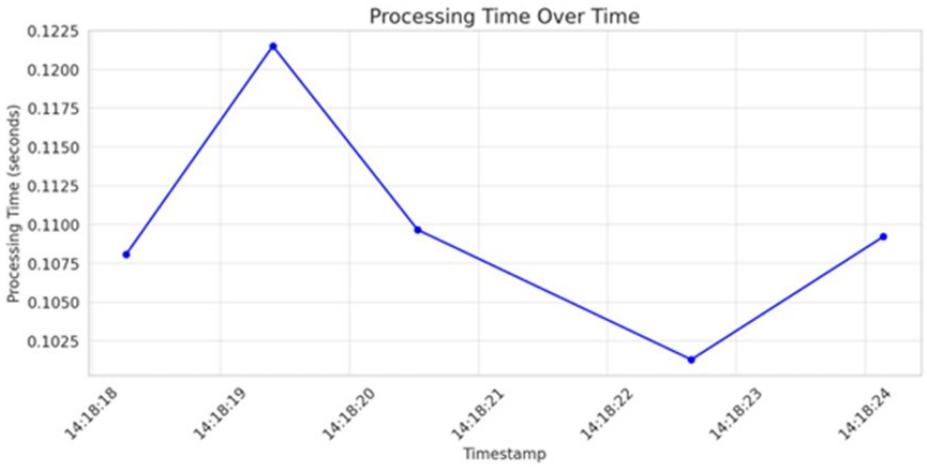

Figure 5.4 : Processing time over time stamps

VI. CONCLUSION

In the proposed work, we have demonstrated an accurate real-time Sign Language Recognition system using a lightweight CNN model in conjunction with OpenCV, MediaPipe, and TensorFlow LITE .The proposed system was successfully implemented on a Raspberry Pi 5, utilizing an external camera for video input and real-time speech output using PYTTSX3 and then audio output via Bluetooth headphones. The processing times varied between 0.102 s and 0.122s which shows that the system is capable of handling real time gesture recognition unlike Transformer systems. All gestures demonstrated confidence scores above 0.965 which showed strong classification performance.The proposed system ensures portability, efficiency, and seamless communication, offering a meaningful step forward in assistive technology for the hearing-impaired community.

VII. FUTURE WORK

This project can be extended to a wider range of gestures and multiple sign languages for the gesture recognition system, thus enhancing the inclusiveness of the tool for users from different language backgrounds. By enhancing the dataset, it's possible to increase the accuracy in varying conditions and different background lightings. This also makes it possible to add compatibility with other sign languages other than Indian Sign Language (ISL).

REFERENCES

[1] N. Adaloglou *et al*., "A Comprehensive Study on Deep Learning-Based Methods for Sign Language Recognition," *IEEE Transactions on Multimedia*, vol. 24, pp. 1750–1762, 2022, doi: 10.1109/TMM.2021.3070438.

[2] S. Shin and W.-Y. Kim, "Skeleton-Based Dynamic Hand Gesture Recognition Using a Part-Based GRU-RNN for Gesture-Based Interface," *IEEE Access*, vol. 8, pp. 50236–50243, 2020, doi: 10.1109/ACCESS.2020.2980128.

[3] K. Wang, L. Huang and H. Liu, "AR Glasses for Sign Language Recognition Based on Deep Learning," in *Proc. 2023 IEEE 5th Int. Conf. Civil Aviation Safety and Information Technology (ICCASIT)*, Dali, China, 2023, pp. 1018–1021, doi: 10.1109/ICCASIT58768.2023.10351656.

[4] S. Srivastava *et al*., "Sign Language Recognition System Using TensorFlow Object Detection API," in *Adv. Netw. Technol. Intell. Comput.*, Cham, Switzerland: Springer, 2021.

[5] C. J. Sruthi and A. Lijiya, "Signet: A Deep Learning Based Indian Sign Language Recognition System," in *Proc. 2019 Int. Conf. Commun. Signal Process. (ICCSP)*, Chennai, India, 2019, pp. 0343–0347.

[6] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif and M. A. Mekhtiche, "Hand Gesture Recognition for Sign Language Using 3DCNN," *IEEE Access*, vol. 8, pp. 79491–79509, 2020, doi: 10.1109/ACCESS.2020.2990434.

[7] J. S., S. N., K. P. V. and S. S. S. G., "Robust Sign Language Recognition Leveraging Bi-LSTM with CNN," in *Proc. 2024 Int. Conf. Emerging Research in Computational Science (ICERCS)*, Coimbatore, India, 2024, pp. 1–6, doi: 10.1109/ICERCS63125.2024.10895323.

[8] D. R. Kothadiya, C. M. Bhatt, T. Saba, A. Rehman and S. A. Bahaj, "SIGNFORMER: DeepVision Transformer for Sign Language Recognition," *IEEE Access*, vol. 11, pp. 4730–4739, 2023, doi: 10.1109/ACCESS.2022.3231130.

[9] Y. Yu, X. Chen, S. Cao, X. Zhang and X. Chen, "Exploration of Chinese Sign Language Recognition Using Wearable Sensors Based on Deep Belief Net," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 5, pp. 1310–1320, May 2020, doi: 10.1109/JBHI.2019.2941535.

[10] S. Sharma and S. Singh, "Vision-Based Sign Language Recognition System: A Comprehensive Review," in *Proc. 2020 Int. Conf. Inventive Comput. Technol. (ICICT)*, Coimbatore, India, 2020, pp. 140–144, doi: 10.1109/ICICT48043.2020.9112409.

[11] N. Basnin, L. Nahar and M. S. Hossain, "An Integrated CNN-LSTM Model for Bangla Lexical Sign Language Recognition," in *Proc. Int. Conf. Trends Comput. Cogn. Eng.*, Singapore: Springer, 2021, pp. 609–617, doi: 10.1007/978-981-33-4673-4_57.

[12] G. M. Rao, C. Sowmya, D. Mamatha, P. A. Sujasri, S. Anitha and R. Alivela, "Sign Language Recognition Using LSTM and MediaPipe," in *Proc. 2023 7th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Madurai, India, 2023, pp. 1086–1091, doi: 10.1109/ICICCS56967.2023.10142638.

[13] A. Gupta *et al*., "Gesture-Based Touchless Operations: Leveraging MediaPipe and OpenCV," *NEU J. Artif. Intell. Internet Things*, vol. 2, no. 1, pp. 30–37, 2023.

[14] J. Bora *et al*., "Real-Time Assamese Sign Language Recognition Using MediaPipe and Deep Learning," *Procedia Comput. Sci.*, vol. 218, pp. 1384–1393, 2023.

[15] J. Donahue *et al*., "End-to-End Adversarial Text-to-Speech," *arXiv preprint*, arXiv:2006.03575, 2020.