International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 9 Issue 3, May-Jun 2025 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

SilentSpeak: AI-Powered Real-Time Sign Language Interpreter

Shamim Ansari, MD. Firdous

Computer Science and Engineering, Institute of Engineering and Management, Kolkata, West Bengal, India

ABSTRACT

This study presents a comprehensive survey of existing hand gesture recognition systems, their advantages and disadvantages, and their applications in various environments. Hand gesture recognition has been a topic of intense interest among researchers in Human-Computer Interaction (HCI) because of the prospects it holds for delivering more natural and intuitive user interfaces. Hand detection and segmentation are key steps involved in gesture recognition since they form the initial stages in movement identification and analysis. These operations can be performed effectively with programming languages like Python, which boasts extensive sets of libraries and tools for computer vision and machine learning. An exemplary application of this technology can be found in the SilentSpeak project, which is an innovative software program aiming to enable communication for those who are nonverbal or speech-impaired. SilentSpeak is an instant sign language interpreter, employing sophisticated computer vision and machine learning techniques to identify, classify, and interpret sign language signs, followed by translating them into written or spoken words. As well as bridging the communication divide between the hearing population and the nonverbal, this ability fosters an inclusive social infrastructure. SilentSpeak's technology architecture is multi-layered. HTML5, CSS, and JavaScript are used to implement the user interface. It is dynamic and interactive. The system's core is a machine learning model that does gesture recognition and is interfaced with the front-end via the Flask web framework in Python. Such a design allows the integration of the model and user interface to be simple, thereby enabling real-time gesture recognition and translation.

How to cite this paper: Shamim Ansari | MD. Firdous "SilentSpeak: AI-Powered Real-Time Sign Language Interpreter" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-9 | Issue-3, June 2025, pp.762-766, URL: www.ijtsrd.com/papers/ijtsrd81066.pdf

IJTSRD81066

KEYWORDS: *Hand Gesture Recognition, Human -Computer Interaction (HCI), Computer Vision, Machine Learning, SilentSpeak, Sign Language Interpreter, Real - Time Translation, Flask Framework*

INTRODUCTION

In the field of technology-enabled inclusiveness, the intersection of computer vision, machine learning, and communication accessibility gave birth to SilentSpeak, an initiative in the form of a software that looks to transform the means used to bridge communication gaps in mute segments of the population. The need for such a technology was critical, given that it responds to the urgency for new technologies allowing sign language users to communicate without difficulty in the largely verbal and written language-oriented environment. As its title would suggest, SilentSpeak is the confluence of sign language interpretation and cutting-edge technology, leveraging the power of high-end computer vision and machine learning algorithms. Its main goal is to develop an in-real-time sign language interpreter that not only accurately identifies and interprets sign language gestures but also offers user-friendly interface that is equally accessible to competent sign language professionals as it is to beginners at this mode of communication. To get an idea of how vital SilentSpeak is, this research study explores the theoretical foundation of hand gesture recognition through the examination of the intricacies sur rounding hand segmentation and detection. This project exceeds technological advancement; it is a total revitalization of accessibility in communication. By exploring this field, SilentSpeak not only proves the potential of technology but also serves as a driving force for social change, facilitating communication among various groups and providing a platform where different forms of communication can all exist together in harmony. This paves the way for a discussion of the theoretical foundations of hand gesture recognition, the intricacy of hand segmentation and detection methods. In discussing

SilentSpeak, it can be seen that this computer project is more than an experimental search for technological innovation; it is a revolutionary attempt with far-reaching implications for inclusive communication. The aim of this study is to document advancements within the field of hand gesture recognition, placing it within the larger vision of SilentSpeak and the role it can play to increase inclusivity for silent people.

RELATED WORK

The early gesture recognition systems were sensor-based and used techniques such as data gloves, inertial sensors, and accelerometers. These sensors picked up exact joint movements and angles and were very accurate. Data Gloves: Hand orientation and finger flexions were correctly captured but in volved users wearing bulky equipment. Motion Sensors: Utilized in detecting hand movement across space, mounted in gloves or armbands. While precise, these systems were costly, invasive, and uncomfortable and therefore impractical for everyday or practical use, particularly in consumer applications. With the advent of deep learning and computer vision, researchers moved towards vision-based gesture recognition, which eliminated the requirement for wearable hardware. The prominent models employed are: Convolutional Neural Networks (CNNs): Powerful for spatial feature extraction from video frames or hand images. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) net works: Employed to capture temporal relationships in sign language sequences. CNN-LSTM Hybrids: Hybrid of CNN for spatial and LSTM for temporal features that performed well on continuous sign recognition. These models yielded encouraging results but were susceptible to needing large datasets, GPU, and significant training time and were thus not suitable for running on low-resource devices such as phones or embedded systems.

Early Wearable Sign Recognition Research: One of the first and seminal works in this area came from Starner and Pentland, who developed wearable sign language recognition systems in the 1990s. Their system employed a head-mounted camera and Hid den Markov Models (HMMs) to recognize American Sign Language (ASL) in real time. This provided the foundation for future work in continuous gesture recognition and context-aware interaction.

Google's MediaPipe platform transformed gesture recognition by providing real-time hand and pose detection using only a regular webcam. Its lightweight architecture relies on pre-trained models for detection of 21 hand landmarks without depth sensors or cameras. MediaPipe is extremely efficient, cross-platform, and simple to integrate, which makes it perfect for user-facing applications on the web and mobile.

Despite advances, many current models suffer from: Heavy computational loads – making them unsuitable for low-power platforms. Lack of user-centric design – too focused on correctness rather than usability or responsiveness in real time. Over-reliance on big data sets – constraining their extrapolation to novel users or personalised gestures.

SilentSpeak overcomes these challenges via the integration of MediaPipe's media landmark detection ability with the light Random Forest classifier. Together, the duo: Minimises resource usage compared to deep learning models. Supports good accuracy for single-gesture recognition. Supports real-time operation on commodity consumer hardware. Emphasises real-world use and fast inference, making it ideal for assistive use and deployment in low-resource settings.

SYSTEM STRUCTURE

Architecture Overview SilentSpeak system comprises:

Input Capture: Webcam stream through OpenCV.

Hand Landmark Detection: MediaPipe detects 21 key points per hand.

Feature Extraction: 42-dimensional flattened vectors (x and y coordinates).

Classification: Random Forest classifier predicts gesture class.

Output Presentation: The resulting label shown on the screen and spoken through pyttsx3.

Dataset Creation:

The dataset contains 4750 samples spread over 19 sign classes with 250 samples per class. Data were collected using MediaPipe from 15 people to preserve variability in terms of hand size, skin tone, and signing style. Each sample records the x and y positions of 21 landmarks.

Training Pipeline:

Features were normalized and split (90% training, 10% test) through stratified sampling. The data were trained on a Random Forest Classifier (scikit-learn). The trained model was saved to a pickle file to be deployed. The classifier was 98.1% accurate, which confirmed robustness and low overfitting.

Real-Time Inference:

In live inference, Media Pipe executes on every frame. For every frame, a prediction is performed and the mode of the last 15 predictions is utilized for stability. Recognized gestures are translated into text and passed on to the speech synthesis module.

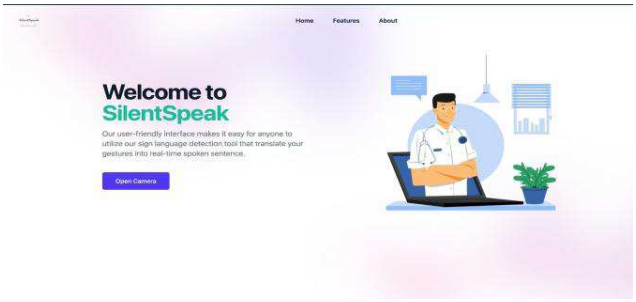

Figure 3.1 Inclusive Design

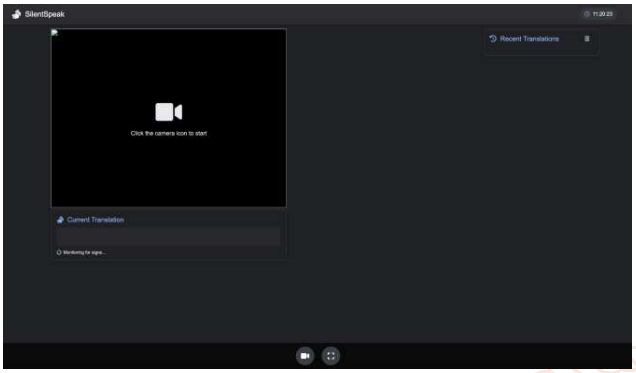

Figure 3.2 Inclusive Design

RESULT AND ANALYSIS

A properly equipped computer configuration is needed to allow the SilentSpeak system hardware installation in order to allow the machine learning model to execute accurately and uninterrupted in real-time. The critical features of the hardware needed are:

A properly equipped computer configuration is needed to allow the SilentSpeak system hardware installation in order to allow the machine learning model to execute accurately and uninterrupted in real-time. The critical features of the hardware needed are:

Running cloud-based services, such as text-to-speech conversion (if hosted remotely).

SOFTWARE COMPONENT

- Python: The primary programming language for implementation.
- Flask (3.0.1): is a web framework designed for server-side application logic. OpenCV (4.7.0.68): is a computer vision library used for image processing.
- Media Pipe (0.9.0.1): A machine learning-based framework for hand tracking and gesture identification.
- Scikit-learn (1.2.1) is a machine learning library used for classification and model training.
- Using known gestures, generative AI creates sentences depending on context.

DATASET PREPARATION

The development of a precise and dependable sign language recognition system, like SilentSpeak, depends heavily on the caliber and organization of the dataset used to train the machine learning model. The dataset preparation was carried out meticulously to ensure it captured diverse, consistent, and meaningful hand gesture data. The process involved the following steps:

Data collection Using Media Pipe Framework: To capture hand gesture photos, we used Google's Media Pipe framework, which is a strong and efficient toolkit for real- time hand tracking and position estimation. Media Pipe provides comprehensive detection and landmark estimation of hands, allowing us to identify major hand landmarks (such as finger tips and joints). Capture photos of gestures with great consistency and accuracy. Extract 21-point 3D landmark coordinates, which were saved alongside each image to improve gesture classification accuracy. Using Media Pipe reduced noise in gesture detection and allowed us to provide high- quality image samples for training deep learning models.

Custom Dataset from Different Individuals: To improve generalizability and reduce model bias, we created a bespoke dataset by gathering gesture photos from 15 distinct people. This step was critical to ensuring: Variability in hand forms, sizes, and skin tones improves the dataset's robustness. Real-world applicability is achieved by taking into account various gesture patterns and signature speeds. Improved performance across several users during the actual deployment. Participants were instructed to perform each gesture in similar lighting and backdrop circumstances to preserve uniformity while allowing for natural variation.

Dataset structure and volume: The dataset was methodically organised to represent 19 unique sign language signals, each of which had:

- 250 images collected from the participants, captured in slightly varying angles, hand positions, and lighting conditions to introduce controlled diversity.
- This results in a total of 4,750 images (19 symbols × 250 images per symbol), providing a balanced and sufficiently large dataset for supervised learning tasks.

FEATURE EXTRACTION

Feature extraction is an important stage in the sign language recognition pipeline because it converts raw visual data into meaningful and concise numerical representations that the machine learning model can easily process. We used the Media Pipe Hands module, a cutting-edge Google framework, for this purpose because of its resilience and real-time performance characteristics.

Removal of Hand Landmarks: The Media Pipe Hands module detects and tracks 21 distinct hand landmarks per frame. These landmarks include key points such as fingertips, joints, and the base of the palm. The module works in real time and is capable of processing both static images and video input from a webcam. It leverages deep learning models internally to precisely estimate the 2D (and optionally 3D) location of each landmark on the hand.

Saving Landmark Coordinates for Analysis:
For each image in the collection, we extracted the (X, Y) coordinate values of every 21-hand landmark. These coordinates were then recorded in a structured manner, such as CSV files or JSON, according to the model training pipeline's needs. Each row of the dataset corresponds to one image and contains:

- The 42 numerical values correspond to the X and Y locations of the 21 landmarks.
- A label denoting the corresponding hand gesture class.
- This structured approach enabled the effective feeding of data into classification algorithms during training.

MODEL SELECTION AND TRAINING:

A. The Random Forest classifier from scikit-learn was used for classification.

B. The MP_Hands model was used to identify 21 predetermined critical locations on the hand.

C. Data preprocessing ensured equal feature lengths by reducing sequences to the minimum possible length.

D. To ensure class balance, the dataset was divided into training (90%) and testing (10%) subgroups by stratified sampling.

E. The model was trained on the processed dataset and evaluated based on accuracy scores.

F. The trained model was preserved in Pickle for future deployment.

DEPLOYMENT AND REAL-TIME GESTURE RECOGNITION:

A. OpenCV was used for real-time image capture and processing.

B. Hand landmarks were recovered from collected frames with Media Pipe.

C. The Random Forest model categorized each gesture.

D. A mode-based technique was used to forecast the final sign after every 15 frames.

E. Google Generative AI (Gemini 2.0 Flash) generated meaningful words from a list of recognized gestures.

F. Recognized gestures were displayed on-screen, with pyttsx3 enabling text-to-speech conversion.

Result and Analysis:
The performance of SilentSpeak was evaluated to identify its capabilities, limitations, and areas for improvement. The following metrics were used for assessment:

Performance Metrics:

A. Accuracy: 98.1%
B. Precision: 0.98
C. Recall: 0.98

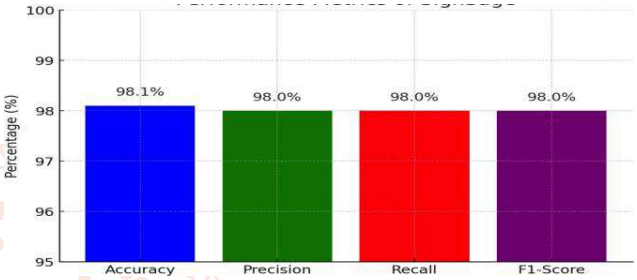

Figure 4.1 Performance Metrics of SilentSpeak

Classification Report

Table 4.1: performance of the system across different gesture labels

Label	Precision	Recall	F1-Score	Support
0	1.0	1.0	1.0	25
1	0.89	0.96	0.92	25
2	1.0	1.0	1.0	25
3	1.0	1.0	1.0	25
4	1.0	1.0	1.0	25
5	1.0	1.0	1.0	25
6	1.0	1.0	1.0	25
7	1.0	1.0	1.0	25
8	1.0	1.0	1.0	25
9	1.0	1.0	1.0	25
11	1,0	1,0	1,0	25
12	0,96	0,88	0,92	25
13	1,0	0,96	0,98	25
14	1,0	1,0	1,0	25
15	0,96	0,92	0,94	25
16	1,0	1,0	1,0	25
17	0,92	0,96	0,94	24
18	0,92	0,96	0,94	25
19	1,0	1,0	1,0	25
20	1,0	1,0	1,0	25

CONCLUSION

A significant advancement in accessible technology, the SilentSpeak project focusses on improving communication for those with speech and hearing impairments. The project successfully demonstrates

how contemporary technologies, when carefully included, can be used to advance inclusion and close the communication gap that frequently separates the hearing population from the mute and deaf community. A strong sign language recognition system at the heart of SilentSpeak uses real-time hand tracking and machine learning techniques to precisely recognise and decipher sign language motions. The system can accurately identify a wide range of hand gestures by training the model on a prepared and varied dataset. The project makes use of a stack of technologies that combines:-

MediaPipe: Accurately collecting the finger positions and hand orientations needed for complex gesture detection with real-time hand tracking and landmark identification.

Scikit-learn: For creating and refining machine learning models, especially for gesture categorisation with hand landmark data that has been retrieved.

Flask structure: To enable users to interact with the system through a browser on a range of devices, develop a small yet effective backend that links the ML model to a web-based interface.

These technologies work together to create a simplified pipeline that takes in input (camera-captured hand movements), processes the data (characteristic extraction and gesture classification), and outputs the results (text or speech). Because it is real-time, this end-to-end solution is perfect for daily communication requirements.

In addition, SilentSpeak gives high priority to user usability and accessibility. The web interface has been designed to allow maximum user friendliness, thereby reducing technical constraints and opening up room for people of different levels of computer literacy. The dedication to real-world usability and social impact by the project is realized through its focus on usability.

In summary, SilentSpeak is an inclusive tool that has the potential to empower marginalized communities, going beyond the scope of being a technological prototype. The tool is a testament to how inclusive innovation and considerate engineering can contribute to more inclusive online communities and greater accessibility in communications.

REFERENCES

[1] Ahmed, F., Paul, P.P., and Gavrilova, M. (2019). Hand gesture recognition using deep learning and media pipe framework. Procedia Computer Science, 152: 701–708.

[2] Oyedotun, O. K., and Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications, 28: 3941–3951.

[3] Molchanov, P., Gupta, S., Kim, K., and Kautz, J. (2015). Hand gesture recognition using 3D convolutional neural networks. In: CVPR, 1–7.

[4] Mittal, N., Singh, R., and Pandey, M. (2020). Real-time Indian Sign Language recognition using deep learning. Procedia Computer Science, 167: 2314–2321.

[5] Zhang, C., Tian, Y., and Liu, Z. (2019). RGB-D-based hand gesture recognition with deep learning. Multimedia Tools and Applications, 78: 30763–30785.

[6] Huang, J., Zhou, W., and Li, H. (2015). Sign language recognition using 3D convolutional neural networks. IEEE ICME, 1–6.

[7] Khan, A.I., Islam, R., and Rahman, M. (2018). A real-time system for sign language recognition using CNN. International Journal of Computer Applications, 179(7): 25–30.

[8] Google Media Pipe Framework (2020). Real-time Hand and Pose Tracking. Retrieved from https://google.github.io/mediapipe/.

[9] Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR).

[10] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. CVPR, 770–778.