Research Article

Sign Voice: Real-Time Sign Language Recognition and Speech Translation System

Dr. R. Salini [1], Jeswin Ebenezer P [2], Karthik R [3], Jason Jacinth Moses A [4], Dr.Jainish G R[5],Dr.P.Deepa[6]

[1,6] Associate Professor, Department Of Computer Science And Engineering Panimalar Engineering College Chennai-600123.
[2] Department Of Computer Science And Engineering Panimalar Engineering College Chennai-600123.
[3] Department Of Computer Science And Engineering Panimalar Engineering College Chennai-600123.
[5] Department of Computer Science And Engineering ,Loyola ICAM college of Engineering and Technology
Corresponding author: [1] shalinirajendran@gmail.com

Abstract

Communication between deaf or hearing-impaired people and the rest of society is still quite a social and technological challenge. The main reason for this is the fact that a very small number of people understand sign language. Although sign language is a very good and expressive way of communicating, the lack of sign language translation tools in real-time is one of the main barriers for inclusive communication in everyday environments. To solve this problem, we have developed SignVoice, a system that recognizes sign language in real-time and converts hands movements into spoken words automatically through computer vision and machine learning. The system we propose exploits MediaPipe to attain the most precise, efficient, and robust hand landmark detection under varied lighting conditions and different backgrounds. A hybrid classification schema has been implemented which allows the system to easily recognize both static and dynamic gestures. Static hand gestures are detected using a K Nearest Neighbors (KNN) classifier, dynamic gestures are analyzed by Long Short-Term Memory (LSTM) networks to understand sequential hand movements based on the change of time. This two-model system increases the recognition accuracy without requiring high computational power. Experimental results demonstrate that the proposed method performs extremely well in terms of recognition accuracy and at the same time demands very little time, thereby establishing the method as highly suitable for assistive communication applications. In summary, SignVoice is a step forward in addressing accessibility issues and promoting social inclusion by enabling communication between hearing-impaired individuals and the rest of the community.

Keywords— Sign Language Recognition, Hand Gesture Recognition, MediaPipe, KNN, LSTM, Computer Vision, Assistive Technology, Real-Time Systems.

I. INTRODUCTION

Millions of hearing-impaired people around the world use sign language as their main method of communication to convey their thoughts, feelings, and information that is essential effectively [8]. However, the general public has very little knowledge of sign language, which causes great communication problems in daily environments like schools, hospitals, government service centers, and work places. Hearing-impaired people often find their accessibility, independence, and social inclusion limited due to this situation. Automated sign language recognition systems offer great potential to resolve this issue as they convert hand gestures into text or speech making it easy for signers and non-signers to communicate without any barriers.

One of the most notable factors that contributed to the current hand tracking/gesture recognition revolution is the computer vision and machine learning field's rapid development. As a result, it has become possible to do these types of applications with an ordinary camera device in real-time. For landmark-based hand modelling and data-driven classification, the research has been intensified and the results show that these two methods are capable of capturing fineness of hand movements with relatively high accuracy. However, the majority of sign language recognition systems that have been proposed so far primarily emphasize on the static gestures, and thus, they do not consider the changing states of dynamic signs which involve continuous motions [6]. Furthermore, some top-performing methods nowadays are based on deep learning architectures that are very efficient in terms of accuracy but extremely demanding in terms of computations. Thus, they require the use of high end GPU hardware as well as large training datasets, which limits the scope of their real-time deployments in resource-limited settings altogether. Particularly dynamic gestures represent a major difficulty because of the variation in the pace of the gestures, the trajectory of the hands, and the personal style of the sign. To be able to accurately identify such gestures, it is necessary to have a very good representation of temporal dependencies, while at the same time having a very low latency allowing real-time interaction [11]. The problem of how to locate the exact point where you can compromise recognition accuracy for computational efficiency is still an open research problem. In this paper, we present SignVoice as a solution to this problem. It is a sign language recognition system that works in real-time and combines traditional machine learning and deep learning approaches to identify both static and dynamic gestures in an efficient way.

The system uses lightweight hand landmark features obtained from video streams, thus cutting down the computational load while still keeping the features that best distinguish the different gestures. The recognition system is a mixture of methods; static gestures are detected by a very fast algorithm in terms of computation, whereas dynamic gestures are trained by a temporal deep learning framework that is able to see the pattern of the motion sequences. The architecture of SignVoice allows high recognition accuracy to be combined with the capability to run in real-time on normal hardware based on CPU, which is a feature that makes it very adaptable to practical assistive communication uses.

II. RELATED WORK

The very first studies on sign language recognition mainly focused on sensor-based systems worn on the body like data gloves, flex sensors, accelerometers, and motion capture devices. Such systems could accurately record the details of fingers joints and hand movements, which made the recognition accuracy very high. On the other hand, the need for special hardware made them intrusive, costly, and not very handy for use throughout the day. Besides, calibration problems and discomfort of the users further reduced the popularity of such systems among the hearing-impaired people who thus turned to vision-based approaches [8].

Later, the sign language recognition systems that are vision-based started to use the typical RGB cameras to capture hand gestures. The first vision-based methods aimed at design features manually such as skin color segmentation, hand contour extraction, edge detection, Hu moments, and shape descriptors. Although these techniques minimized hardware dependency, the performance of the system was very sensitive to environmental factors such as the light conditions, the background clutter, and the skin color variations. Besides, the handcrafted features were often unable to generalize different users and complex gesture sets.

The rise of deep learning has led to significant breakthroughs in sign language recognition. For example, Convolutional Neural Networks (CNNs) have been used extensively to automatically extract spatial features from images of hands, thus getting rid of the need for manual feature engineering.

Table 1. Comparison with Existing Systems

Feature	Existing Systems	Proposed System (SignVoice)
Hardware Requirement	Sensor gloves	Webcam only
Real-Time Processing	Limited	Yes
Gesture Types	Mostly static	Static + Dynamic
Cost	High	Low
Accessibility	Limited	High

In the case of dynamic gestures, Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) cells have been employed to learn temporal relations in different video frames. The CNNLSTM combined models have achieved remarkable recognition accuracies for both continuous and isolated sign language recognition [11]. Nevertheless, these deep-learning-based methods generally require massive annotated data, a lot of training time, and powerful GPU machines, which imposes a real challenge to the live implementation of these methods on low-cost or CPU, only devices.

Figure 1. Sample of already existing datasets.

There are quite a few studies revealing that a combination of landmark-based features and lightweight machine learning classifiers can be a winning formula not only in terms of the recognition accuracy but also the computational cost saving [3], [4], [12]. These types of methods have handy compact feature representations, which are less prone to be influenced by the background changes or the illumination conditions. However, a majority of the current landmark-based systems are inherently limited because they either focus solely on static gestures or dynamic gestures, thus making them less versatile and adaptable to real life communication situations involving both pose-based and motion-based signs.

The suggested SignVoice system is based on this work and takes it one step further by developing a hybrid recognition framework that merges classical machine learning methods for static gesture classification and deep-learning-based temporal modelling for dynamic gesture recognition. The authors have effectively dealt with both accuracy and real-time performance issues by fusing a computationally efficient spatial hand pose classifier [1] with a temporal learning model that can capture sequential motion patterns.
This well, thought out combination makes SignVoice stand out from other landmark-based systems and allows running the software smoothly on standard CPU-based hardware.

III. SYSTEM ARCHITECTURE

The whole setup of the proposed SignVoice system is shown in Fig. 2. The system is set up in a modular way, as a pipeline for processing in real-time, with each module doing a particular job and communicating smoothly with the next components. Modular architectures like this are very common in real-time computer vision systems because they offer better scalability, maintainability, and make it easier to add new features or upgrade. The system takes video frames continuously from a regular webcam and turns the recognized hand gestures into the equivalent speech output with very little delay, thus allowing smooth real-time human computer interaction [8].

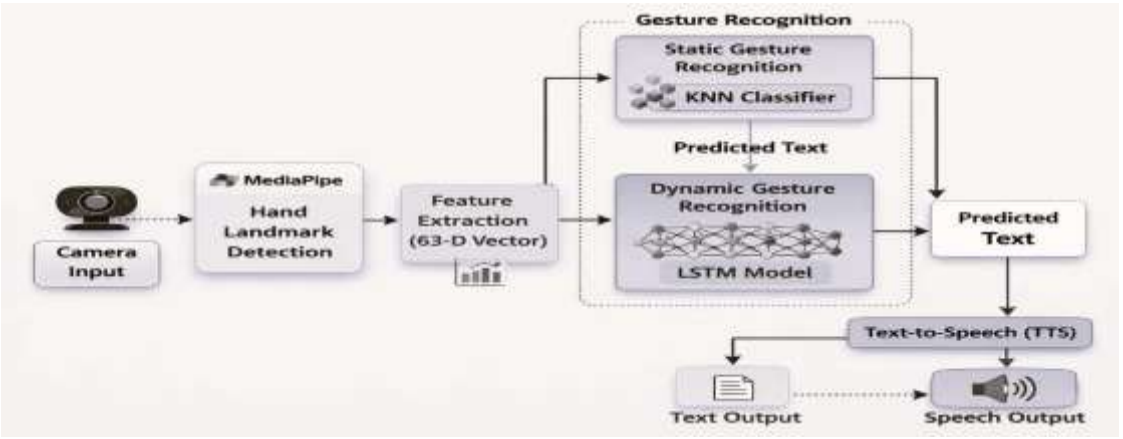


Figure 2. System Architecture.

A. Camera Input Module

The camera input module's job is to take live videos through a normal webcam. To keep the flow of time smooth, which is very important for recognizing changes in gestures correctly [7], frames are grabbed at a constant frame rate. Each frame is then reshaped and normalized before going to the hand landmark detection module, thus lessening the computational load and guaranteeing uniform input sizes. This module is the main gateway connecting the real-world with the digital processing pipeline, a method widely used in real-time vision-based gesture recognition systems [10].

B. Hand Landmark Detection Module

This module detects and tracks hand regions in every video frame by utilizing the MediaPipe hand tracking framework. MediaPipe determines 21 key landmarks for each hand, which correspond to joints and fingertip locations. The landmark-based representation greatly diminishes visual complexity and at the same time retains the necessary spatial and geometric information. The detected landmarks, due to the great architecture of the system, can be trusted even when the background is changed, the light is different, or the hands are at different angles, thus making the system appropriate for real-life use [3].

C. Feature Extraction Module

During feature extraction, the hand landmarks identified are changed into numeric feature vectors. The (x, y, z) coordinates of every point are taken out and normalized referring to the wrist joint so that the features become scale and translation invariant [4]. The normalization makes the system more reliable to different hand sizes and distances from the camera. The feature vectors thus obtained offer a concise and distinguishing representation of hand gesture and movement, which can be effectively classified with very little computing power.

D. Gesture Classification Module

The gesture classification module uses a hybrid recognition strategy to distinguish effectively between static and dynamic gestures. Static gestures are classified with a K Nearest Neighbors (KNN) algorithm that is known to give high performance on low-dimensional landmark-based feature representations and also enables fast inference without the need for a large training set [1]. On the other hand, dynamic gestures whose main characteristic is the temporal motion patterns, are classified by a Long Short Term Memory (LSTM) network which is capable of modelling the sequential dependencies of multiple frames. The hybrid method thus provides a good compromise between the recognition accuracy and real-time performance and at the same time keeps the computational complexity at the minimum level.

E. Speech Output Module

Once a gesture has been identified, the associated textual label is sent to the speech output module. A text-to-speech converter changes the recognized gesture into spoken words, thus giving the user instant feedback and allowing for a more natural interaction [8]. This module makes the whole assistive communication pipeline from end to end and thus communication between hearing-impaired people and non-signers in real-world environments becomes possible and effective.

Table 2 System Modules

Module Name	Description
Camera Input Module	Captures real-time hand gestures using webcam
Hand Landmark Detection Module	Detects hand landmarks using MediaPipe
Feature Extraction Module	Converts landmarks into numerical feature vectors

Gesture Classification Module	Classifies gestures using KNN and LSTM models
Speech Output Module	Converts recognized text into speech

IV. DATA ACQUISITION AND PREPROCESSING

Accurate gesture recognition is very much dependent on the quality of the dataset and the effectiveness of preprocessing techniques, because these factors directly affect the feature's reliability and classification performance.

Thus, the proposed system (Fig 3) uses two different datasets to tackle static and dynamic gestures separately, which allows the use of specialized modeling strategies for spatial hand poses and temporal motion patterns, respectively. This division not only boosts the recognition performance but also makes the model design simpler and the computational load for a real-time application lower.

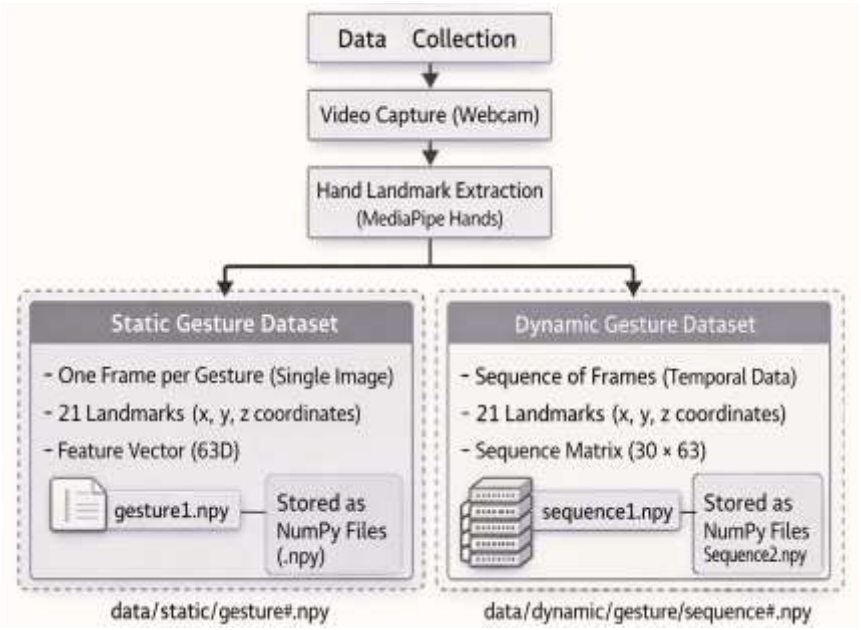


Figure 3. Dataset Architecture.

4.1 Static Gesture Dataset

Static gestures represent hand poses which can be recognized just from one frame of the video without any temporal context [12].

Each static gesture instance is recorded with the help of a regular webcam in a controlled environment and then processed via the MediaPipe pipeline to get the hand landmarks.

The (x, y, z) positions of the 21 identified landmarks are concatenated into a 63, dimensional feature vector (21 landmarks 3 coordinates), thus giving a compact yet discriminative representation of a hand posture.

The features thus obtained are saved in NumPy format for fast loading, preprocessing, and computation during both the training and inference stages.

Dataset structure:

data/static/gesture_name/sample.npy

Each file corresponds to a single static gesture instance. To boost the robustness, several samples are taken for each gesture that include variations in hand orientation, distance from the camera, and different ways of signing by individuals.

4.2 Dynamic Gesture Dataset

Unlike static gestures, which can be recognized using just a single video frame, dynamic gestures consist of hand movements that change continuously over time and therefore cannot be recognized by a single frame . To record

each dynamic gesture, a sequence of frames is taken over a set time period in order to capture the changes happening over time.

Therefore, each frame of the sequence is used for extracting the hand landmarks with a stable detection framework, and hence a time-ordered series of feature vectors that can be used for temporal modeling is obtained [11].

Such a sequential representation permits the modeling of motion patterns and learning of temporal dependency

through the dynamic sign language gestures.

Figure 4. Sample of Dynamic Gesture Dataset.

Each dynamic gesture sample (Fig 4) is a matrix, where the rows are time steps and the columns are features based on landmarks, thus providing a structured temporal representation [2].

This temporal structure makes it possible for the LSTM network to accurately understand the motion patterns and sequential dependencies of dynamic gestures.

Dataset structure:
data/dynamic/gesture_name/sequence.npy Preprocessing mainly consists of normalizing the length of sequences by padding or cutting, normalizing the features, and filtering the noise to have the consistent input dimensions for different samples. Such preprocessing methods help to stabilize the models, make them less sensitive to variations in executions, and generally improve the recognition accuracy of dynamic gesture classification tasks.

V. FEATURE EXTRACTION USING MEDIAPIPE

The proposed system leverages MediaPipe Hands for precise and Real-time detection of hand landmarks [3]. MediaPipe Hands locates 21 key points of the human hand that are closest to anatomy, such as the wrist, finger joints, and fingertips, thus facilitating a highly accurate modeling of hand posture and movement.

Each landmark is depicted in a 3D coordinate space by means of (x, y, z) values, where (x, y) stand for normalized image coordinates and z denotes relative depth information. Such a representation is a minimal yet highly descriptive form of hand geometry which is well, suited to real-time gesture recognition applications [3].

Figure 5. 21 Hand landmarks.

The feature vector is defined as:

$$\mathbf{F} = [x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_{21}, y_{21}, z_{21}]$$

>(1) --------

For each hand detected a feature vector is generated by merging into one vector all the coordinates of the detected landmarks [3]. Features are:

A. Number of landmarks: 21
B. Coordinates per landmark: 3 (x, y, z)
C. Total feature dimension: 21 × 3 = 63

To avoid any issues of landmark coordinates being different from one sample to another, the coordinates of the landmarks are normalized to the wrist joint, thus giving translation and scale invariance [4]. This normalization allows the system to accommodate the changes in hand sizes and distance from the camera without any difficulty, thus becoming more reliable when different users and capture conditions are involved [12].

The extracted landmark-based features have several advantages:

1. Robustness to scale variations, enabling reliable detection at different camera distances.
2. Reduced sensitivity to background noise, as only skeletal hand information is processed.
3. Computational efficiency, making the features suitable for real-time applications on CPU-based systems.

Such a compact yet descriptive encoding becomes the basis of both static and dynamic gesture recognition, thus providing the classification with high accuracy while the computational complexity remains minimal [1], [2].

VI. STATIC GESTURE RECOGNITION USING KNN

Static gesture recognition mainly deals with hand poses which can be easily recognized from one video frame only without the need for any temporal context. A K Nearest Neighbors (KNN) algorithm is used for this mission. KNN is non parametric and instance-based classifier. This makes it highly suitable for the use of low dimensional feature spaces such as landmark-based hand representations [1]. The below Fig 6 shows the latency variation of the Sign Voice system.

Being a simple algorithm with minimum requirement for training KNN makes it possible to have a quick and reliable classification of static hand gestures on a live video feed.

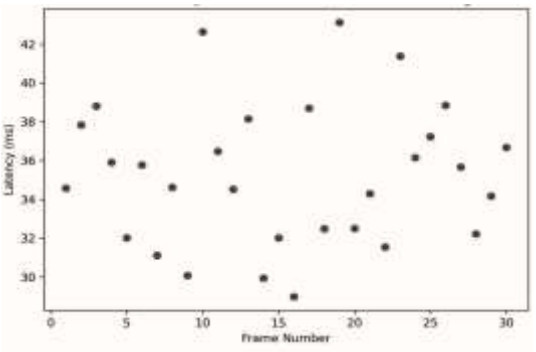


Figure 6. Sample graphs of latency variation.

6.1 Distance Metric

The degree of similarity between two samples of static gestures is estimated by the Euclidean distance between their respective 63-dimensional feature vectors, which is a typical measure for nearest neighbor-based classification [1]. This distance metric accurately reflects the variations in geometry of hand poses in the landmark feature space while being computationally cheap, thus it is very suitable for real-time static gesture recognition applications.

The Euclidean distance that separates two feature vectors is given by the formula:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{63} (x_i - y_i)^2}$$

---------- >(2)

6.2 Classification Rule

K Nearest Neighbors (KNN) algorithm finds the k closest samples in the training dataset to a test feature vector

using Euclidean distance [1]. The class label is predicted by majority voting among these nearest neighbors. The value of k is empirically chosen to provide a good trade-off between robustness to noise and classification sensitivity, which is a common practice in nearest-neighbor-based classification methods.

KNN is chosen for the recognition of static gestures for the following reasons:

1. Simple and easy to implement Inference time is very fast.
2. Since there is no need to train a complex model or optimize parameters, good performance on distinctly hand poses features,e.g., landmark-based representations.

Due to these features, KNN is quite efficient for usage in the Real-time recognition of static gestures on low dimensional feature spaces more so in CPU based systems.
For a test sample x, the class prediction is made by:

$$\hat{c} = \arg\max_c \sum_{i \in N_k} I(y_i = c)$$

>(3) --------

where N denotes the collection of the k closest neighbors.
KNN has been the selected method owing to its straightforward nature, quick prediction, and adequate capability for static hand poses.

VII. DYNAMIC GESTURE RECOGNITION USING LSTM

Dynamic gestures require the hands to be in motion for a number of video frames and thus, they have temporal dependencies that cannot be sufficiently captured by single frame classifiers only. To solve this issue, the paper system first uses an LSTM (long short term memory) network, which is a recurrent neural network of a special kind that was made for sequence data modelling and also for learning long-term dependencies [2].

7.1 Input Representation

Each dynamic gesture is encoded as a series of feature vectors derived from landmarks, with each time step equating to one video frame. In other words, a gesture sequence is made up of several 63, dimensional vectors that are time-ordered and reflect the continuous changes of the hand movement. In order to maintain consistency between different samples and to facilitate batch processing gesture sequences are either padded or cut to a predetermined length. This is a standard practice in the preprocessing of sequence-based deep learning models.
Each dynamic gesture is represented as a sequence:

$$\mathbf{X} = \{x_1, x_2, \ldots, x_{30}\}, \quad x_t \in \mathbb{R}^{63}$$

---------- >(4)

7.2 LSTM Model

While processing the input sequence, the LSTM network goes through it frame by frame and also keeps track of the internal memory state which is updated at every time step [2]. By means of its gates the input gate, forget gate, and output gate the LSTM selectively chooses to keep or throw out information. In this way, it is able to understand long-term temporal dependencies in sequential data. So, the model is able to recognize complex motion patterns like hand trajectories and changes in directions that are typical of dynamic gestures [7].
The LSTM's last hidden state is the output of the dense layer fed into the softmax layer that gives class probabilities for gesture classification. This model design allows the model to recognize gestures that look alike in the single frames but are quite different in the sequence of movements, which is a feature that aids dynamic sign language gesture recognition accuracy.

VIII. HYBRID RECOGNITION STRATEGY

The suggested system is a hybrid recognition one which utilizes the advantages of both classical machine learning and deep learning methods (Fig 7) in order to effectively deal with various kinds of sign language gestures. In this framework:

A. Static gestures are recognized by using the K Nearest Neighbors (KNN) classifier, which is very suitable for pose-based recognition with low dimensional landmark feature vectors and allows fast inference with very low computational overhead [12].
B. Dynamic gestures are recognized by a Long Short Term Memory (LSTM) network,

which can capture temporal dependencies and learn motion patterns over sequences of frames [2].

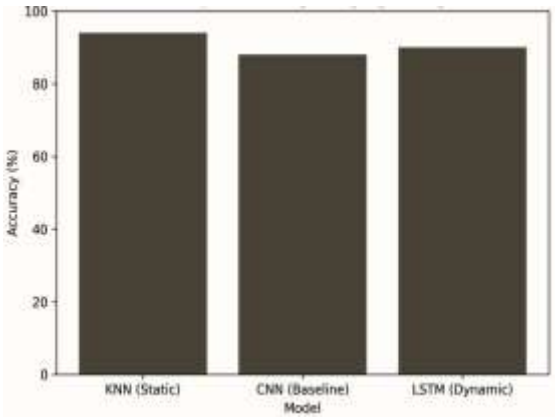


Figure 7. Comparison graph of KNN, CNN and LSTM

This division of labor brings about several important advantages:

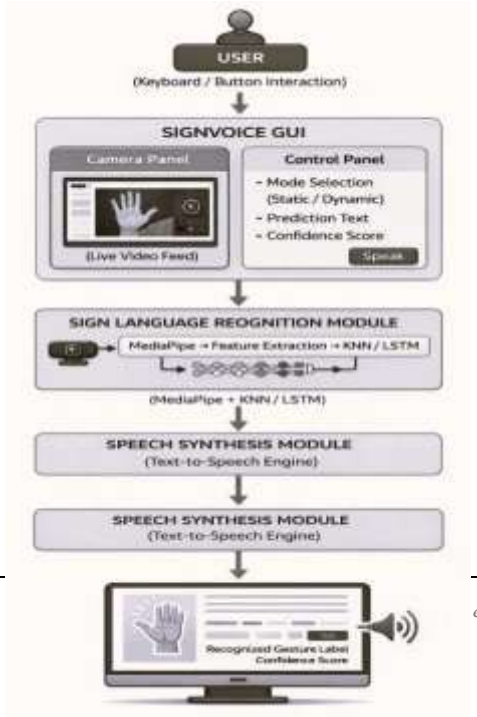
1. Lower training complexity since static gesture recognition is done with a very simple algorithm and hence deep neural networks are not necessary for them.
2. Faster speeds during Real-time inference especially for the statics which are used most frequently because nearest neighbor classification has very low computational cost.
3. Better overall recognition accuracy by limiting the use of deep temporal models to just those instances when there is motion present.
4. Greatest possible utilization of computational resources allowing the system to be deployed confidently on standard CPU only hardware without GPU acceleration.

By some clever combination of lightweight spatial learning and heavy temporal deep learning techniques, the proposed hybrid architecture practically finds the right spot between performance, accuracy, and real-time usability thus being quite suitable for use as an assistive sign language communication system in everyday environments.

IX. GRAPHICAL USER INTERFACE

A Fullscreen desktop graphical user interface (GUI) has been built with PyQt to effectively facilitate and aesthetically offer an easy, friendly, and beautiful interaction platform to the SignVoice system. The GUI can operate live and at the same time be clear and easy enough for both the hearing-impaired users and non-signers to understand. Usually, Real-time onscreen visual interfaces run on computer vision-based gesture recognition systems to make human computer interaction more attractive and the system more responsive quite in line with the findings of the authors in [5], [10]. Below (Fig 8) shows the GUI Architecture of the SignVoice System.

Considerable work has been put into the visual layout and interface responsiveness in order to avoid the screen of the user being cluttered with unnecessary information/visuals which can detract the user during live gesture



recognition hence leading to an improved user experience and higher interaction efficiency in assistive communication applications [8].

Figure 8. GUI Architecture.

The GUI features the main elements as follows:

A. **Live camera feed:** Shows the Real-time video from the webcam to offer the user immediate visual feedback.
B. **Gesture prediction display:** The recognized gesture label is clearly shown to the user for easy understanding.
C. **Confidence score visualization:** Indicates the trustworthiness level of the classifier thus it is more transparent and credible to the user.
D. **Mode switching controls:** Users can easily switch between static and dynamic gesture recognition modes.
E. **Speech output trigger:** Allows immediate text to speech conversion of recognized gestures.

These interface components are typical for gesture recognition in real-time and assistive systems to enhance interaction clearness and accessibility. The GUI layout prioritizes prediction outputs and confidence indicators to be very visible without covering the live camera feed. Such a design decision improves the user experience during demos, learning situations, and real-life assistive communication applications, where clarity and quick response are paramount.

X. EXPERIMENTAL RESULTS

Recognition accuracy, robustness, and real-time performance of the proposed system is evaluated through gesture datasets of both static and dynamic types. To show the practicability of the proposed method in resource, poor environments, all the experiments are performed on standard CPU-based hardware without GPU acceleration.

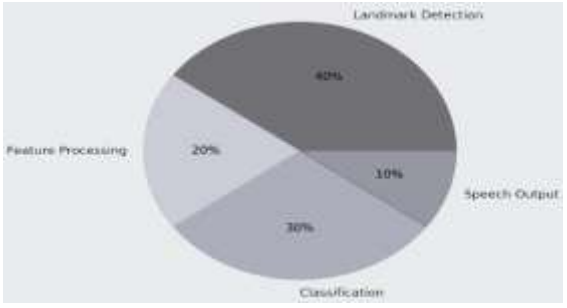


Figure 9. Processing Distribution of Sign Voice System

The Fig 9 shows the processing time distribution in the SignVoice pipeline. Landmark Detection takes the largest portion of time at 40%, making it the most computational step. Classification accounts for 30%, while Feature Processing uses 20% of the total processing time. The smallest portion is Speech Output, which requires only 10% of the overall processing time.

10.1 Static Gesture Performance

The static gesture recognition module records an average accuracy of about 94% over several test gestures. The KNN-based classifier exhibits a steady and consistent performance in typical indoor lighting, which further validates the gesture recognition tasks based on pose as a good use case for it [1]. Because of the discriminative power of the landmark-based features, the static gestures of different hand poses can be identified with a very high level of reliability and with a very low level of classification confusion.

Figure 10. Sample output of Static Gestures

10.2 Dynamic Gesture Performance

It is shown that the dynamic gesture recognition module has an average accuracy of around 90%.
The LSTM network is able to extract temporal motion patterns from the gesture sequences and hence correctly differentiate even those gestures that may have similar intermediate poses but different overall motion trajectories [2].

Most of the time, slight confusion between the two classes results in mixed, up gestures that share a part of the motion or whose execution speeds vary. This is however the main disadvantage of any dynamic gesture recognition system based on temporal modeling.

Figure 11. Sample output of Dynamic Gestures.

10.3 Real Time Performance

The proposed system shows very good real-time performance features when running on ordinary CPU-based hardware:

A. **Frame rate:** 25–30 frames per second
B. **Average latency:** approximately 35 milliseconds
C. **Execution environment:** CPU-based system

The findings show that the hybrid framework meets the time constraints for interactive gesture recognition systems, and it does not need to be supported by GPU acceleration. The performance measured agrees well with that of previous landmark-based and lightweight vision systems for real-time human computer interaction [10].

Table 3. Performance Table

Model	Gesture Type	Accuracy
KNN	Static Gestures	92%
LSTM	Dynamic Gestures	90%
Overall System	Combined Gestures	91%

The Table-3 measures how correctly the system identifies the performed gestures. The accuracy depends on factors such as lighting conditions, hand orientation, gesture clarity, and dataset quality.

XI. DISCUSSION

The experimental evaluation reveals that landmark-based feature extraction combined with a hybrid classification approach is an optimal solution balancing recognition accuracy and computational efficiency. Lightweight hand landmarks have drastically lowered input dimensionality, whereas using deep learning only for the dynamic gestures has helped in avoiding unnecessary computational costs. However, the system might not

work as well in uncontrolled environments with poor lighting, occlusion of hands, or highly varied ways of performing gestures. Besides that, the present system is only capable of recognizing isolated gestures which means that it might not be very useful in a continuous conversation. Still, the evidence shows that the proposed method is practical and robust enough for real-time assistive use.

XII. FUTURE WORK

Future enhancements to SignVoice system can be:

A. Perform continuous sentence, level sign language recognition Using NLP techniques for grammar correction and sentence refinement.
B. Expanding the gesture vocabulary to cover more language Supporting two hand and bimanual gestures.
C. Porting on mobile and embedded platforms for portability.

These additions are expected to not only increase the expressiveness of the system but also its ease of use and accessibility, making the system real conversational sign language translation.

XIII. CONCLUSION

This paper discusses the sign language recognition and speech translation system with a hybrid machine learning framework. The system combines MediaPipe, based hand landmark extraction with KNN for static gesture recognition and LSTM for dynamic gesture modeling, thus resulting in accurate, efficient, and low-latency performance. The proposed method is highly effective even on a conventional CPU and offers a user-friendly graphical interface, paving the way for real-world assistive communication applications and greater social inclusion of hearing-impaired persons. The system maintains robust recognition across diverse camera angles and hand scales without specialized hardware.

XIV. REFERENCES

[1] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, Jan.1967.

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

[3] Google Research, "MediaPipe Hands: On-Device Real-Time Hand Tracking," 2021. [Online]. Available: https://developers.google.com/mediapipe

[4] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[5] J. Shotton et al., "Real-Time Human Pose Recognition in Parts from Single Depth Images," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1297–1304.

[6] A. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand Gesture Recognition with 3D Convolutional Neural Networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 1–7.

[7] P. Molchanov, X. Yang, S. Gupta, K. Kim,
S. Tyree, and J. Kautz, "Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4207–4215.

[8] R. Rastgoo, K. Kiani, and S. Escalera, "Hand Sign Language Recognition Using Deep Learning," Expert Systems with Applications, vol. 164, 2021.

[9] M. Abavisani, H. Patel, and V. M. Patel, "Improving the Performance of Unimodal Dynamic Hand-Gesture Recognition with Multimodal Training," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 1–14, Apr. 2020.

[10] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[11] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A Modified LSTM Model for Continuous Sign Language Recognition," Pattern Recognition Letters, vol. 142, pp. 282–289, 2021.

[12] J. Kaur and A. Kaur, "Vision-Based Hand Gesture Recognition for Human–Computer Interaction: A Review," International Journal of Computer Applications, vol. 181, no. 46, pp. 1–8, 2018.