Sign Language Recognition Using Machine Learning

Anuja Suresh Vishe
MSc Information Technology
D. G. Ruparel College Mumbai, Maharashtra, India
Academic Year: 2025–2026

ABSTRACT

Many individuals who use sign language face a lot of difficulties in communicating and learning even translating it usually takes a lot of time and requires interpreters. This research introduces SignSpeak a system that recognizes sign language in real time using machine learning. It uses Media Pipe Hands to detect hand movements and a random forest model to identify gestures.the system adjusts hand data to work for different hand sizes and can recognize signs like Hello, yes, No, Okay,please and thank you accurately streamlit web interface show a live webcam feed with the detected gestures and confidence scores. Tests show that SignSpeak can quickly and correctly recognize hand signs, helping people communicate more easily

KEYWORDS

Sign language recognition, Hand Gesture recognition, MediaPipe, Random Forest, Machine learning, Human-computer Interaction

1.INTRODUCTION

Sign Language is an important way of communication for people who are deaf or hard of hearing. However, many people do not understand sign language and this creates a communication gap. Often, interpreters or special training are needed which may not always be available. Automatic sign language recognition can help to solve this problems by translating signs into text or speech in real time [1].

With recent improvements in machine learning and computer vision, it has become easier to track hand movements using a camera. Tools like MediaPipe Hands can detect important points on the hand accurately [7]. These hand points can then be used by models to recognize different gestures. Even with these advancements, challenges such as changes in lighting, background, and the need for fast and accurate results still exist [2], [3].

This study presents the SignSpeak machine learning system designed to recognize sign language gestures in real time. The system uses a webcam to record hand movements and then identify key points on the hand. These hand landmarks are adjusted to minimize differences caused by hand size or distance from the camera. A Random Forest model is then used to recognize the performed sign [4], [6]. The predicted output is displayed instantly on a Streamlit interface, along with a confidence score for better understanding.

2.LITERATURE REVIEW

Sign language recognition has progressed considerably to improve communication accessibility for individuals with hearing and speech impairments. Early studies mainly reviewed and categorized recognition techniques, highlighting challenges such as limited datasets, gesture variability, and real-time constraints [1]. Initial systems relied on sensor-based methods, including wearable gloves and specialized hardware, which provided accurate tracking but were costly, uncomfortable, and impractical for real-world use [2], [3].The shift toward vision-based recognition using cameras removed the need for external sensors and improved usability [2]. However, these methods were often affected by lighting conditions, background noise, and hand occlusion [3]. With the adoption of deep learning, particularly CNN-based models, recognition accuracy improved significantly in real-time systems using webcam input [4], [6]. Despite their effectiveness, such models require large datasets and high computational resources.To address these limitations, landmark-based approaches using frameworks like MediaPipe were introduced. By extracting key hand points, these methods reduced data complexity while maintaining reliable gesture representation [7]. When combined with machine learning classifiers, they enabled faster and more efficient real-time recognition. Although sequence-based models and hybrid deep learning approaches further improved performance for dynamic gestures [8], [9], they remained computationally intensive.To overcome these challenges, the proposed SignSpeak system employs a lightweight hand landmark-based approach with a Random Forest classifier, ensuring efficient real-time performance with low computational cost and practical deployability.

3.PROPOSED SYSTEM ARCHITECTURE

The proposed SignSpeak system is designed using a simple client–server approach. It consists of three main modules: data collection, model training, and real-time prediction. These modules work together to recognize sign language gestures efficiently [4], [6].

3.1 System Overview

Input:
A live video feed is captured using a webcam to record hand gestures [4].

Processing:
The MediaPipe framework is used to detect the hand and extract 21 important hand landmarks from each video frame [7].

Feature-Engineering:
The extracted landmarks are normalized using the min–max scaling method to reduce differences caused by hand size and position [3].

Classification:
A Random Forest classifier analyzes the processed landmarks and predicts the corresponding sign language gesture [4], [6].

Output:
The final result is displayed on a Streamlit dashboard, showing the live video along with the predicted gesture and its confidence score [4].

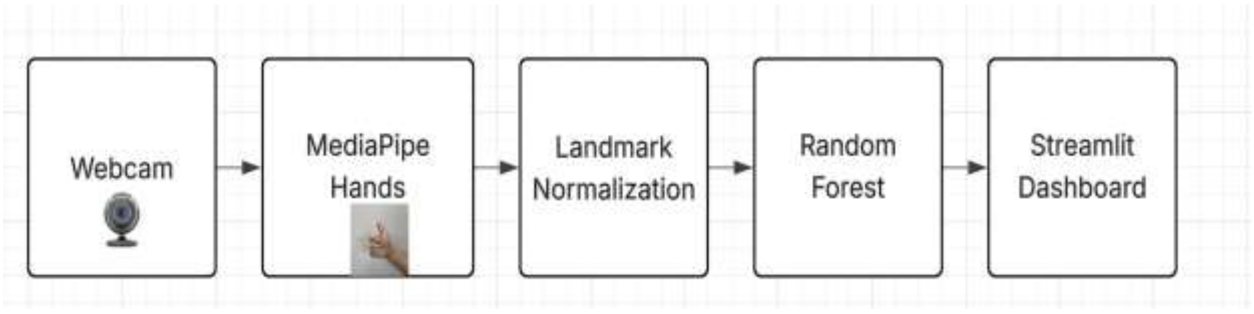


Fig. 1: Proposed SignSpeak System Architecture

3.2 Data Collection Module

The data collection module is responsible for acquiring hand gesture data in real time using a webcam [4], [6]. For each video frame, the system detects the hand and extracts 21 landmark points using the MediaPipe Hands framework [7]. Each landmark is represented by its two-dimensional (x, y) coordinates, capturing the spatial structure of the hand.

The dataset includes commonly used sign language gestures such as Hello, Yes, No, Okay, Please, and Thank You. For every gesture instance, the extracted landmark coordinates are labeled and stored in a CSV file, which serves as the input for model training and evaluation [6].

To reduce variations caused by differences in hand size, position, and distance from the camera, the landmark coordinates are normalized using min–max scaling [3]. The normalization is defined as:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}, y_{norm} = \frac{y - y_{min}}{y_{max} - y_{min}}$$

where x_{min}, x_{max}, y_{min}, and y_{max}represent the minimum and maximum coordinate values of the detected hand landmarks in a given frame.

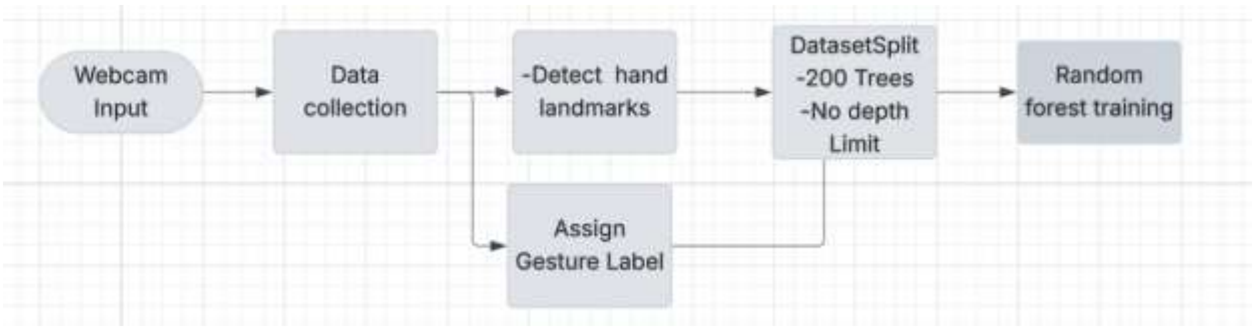


Fig. 2 – Data Collection Workflow

3.3 Model Training Module

The system uses a Random Forest classifier to train the sign language recognition model [4], [6]. The collected dataset is divided into two parts: training and testing, ensuring that all gesture classes are equally represented [4].

The Random Forest model is trained using 200 decision trees, which helps improve prediction accuracy. No limit is set on the tree depth so the model can learn useful patterns from the data. The performance of the trained model is measured using accuracy [4], [6].

Random Forest is chosen because it can handle complex gesture patterns, reduces overfitting, and provides reliable results [1]. After training, the model is saved using the joblib library so it can be reused later for real-time gesture prediction [6].

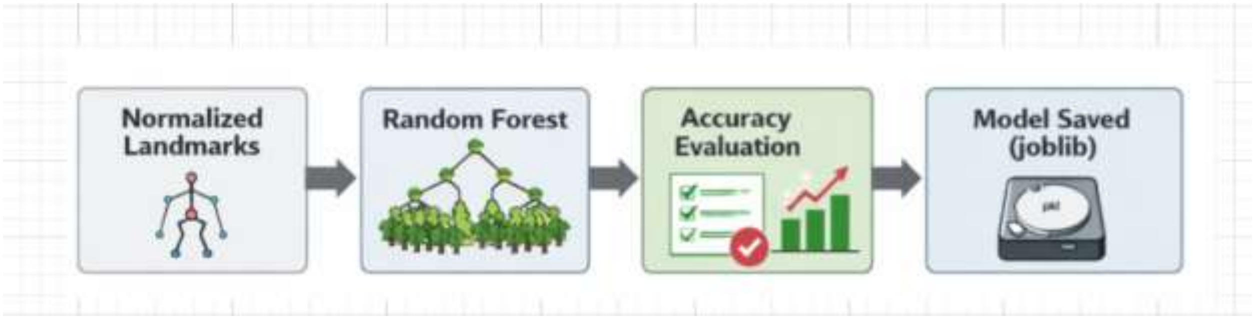


Fig. 3: Model Training Pipeline

3.4 Real-Time Prediction Module

The real-time prediction module is responsible for recognizing sign language gestures from live video [4], [6]. The system captures video frames from the webcam and flips them horizontally to provide a natural viewing experience for the user [4].

For each frame, the MediaPipe framework is used to detect the hand and extract its landmark points [7]. These landmark coordinates are then normalized to values between 0 and 1 to maintain consistency with the training data [3], [4].

The trained Random Forest model processes the normalized landmarks and produces a probability score for each gesture [4], [6]. A gesture is displayed only when the prediction confidence is 60% or higher, which helps reduce incorrect predictions [4].

The results are presented through a Streamlit dashboard, which shows the live webcam feed along with the predicted gesture label and its confidence percentage in real time [4].

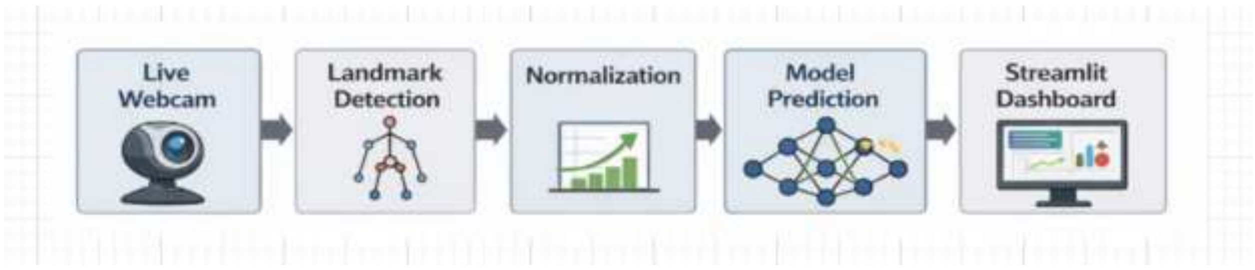


Fig. 4: Real-Time Prediction Flow

4. METHODOLOGY

The methodology followed in this research consists of several structured steps to build and deploy the sign language recognition system.

Step 1: Data Collection
Hand gesture data is collected using a webcam. For each sign, hand landmarks are detected and multiple samples are recorded. All collected data is stored in a CSV file along with the corresponding gesture labels [1][2].

Step 2: Preprocessing
The collected data is cleaned to handle any missing or incorrect values. The hand landmark coordinates are then normalized to reduce the effects of hand size, position, and distance from the camera [2][3].

Step 3: Feature Engineering
Each hand gesture contains 21 landmarks, with x and y coordinates for each point. These landmarks are converted into a single 42-element feature vector, which is suitable for input into the machine learning model [3][4].

Step 4: Model Training
A Random Forest classifier is trained using the preprocessed feature vectors. The model's performance is evaluated using a separate test dataset to measure its accuracy [4][5].

Step 5: Real-Time Deployment
The trained model is integrated into a Streamlit-based application. Live webcam frames are processed to extract hand landmarks, predict the gesture, and display the gesture label along with its confidence score in real time [4][5].

5. SYSTEM IMPLEMENTATION AND USER INTERFACE

The SignSpeak system is built using Python. MediaPipe detects hand landmarks, OpenCV handles webcam input[3], and scikit-learn runs the Random Forest classifier. The Streamlit interface provides a simple, user-friendly platform for real-time gesture recognition with live visual feedback.

5.1 Home Page Interface

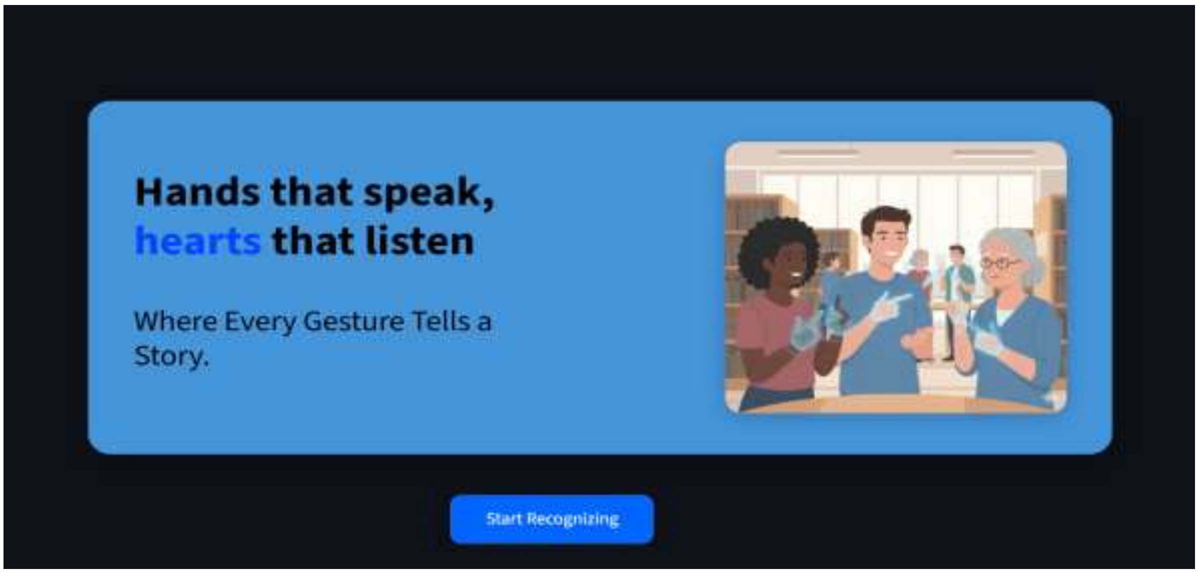


Fig. 5: Home page of the SignSpeak application

5.2 Live Gesture Recognition Interface

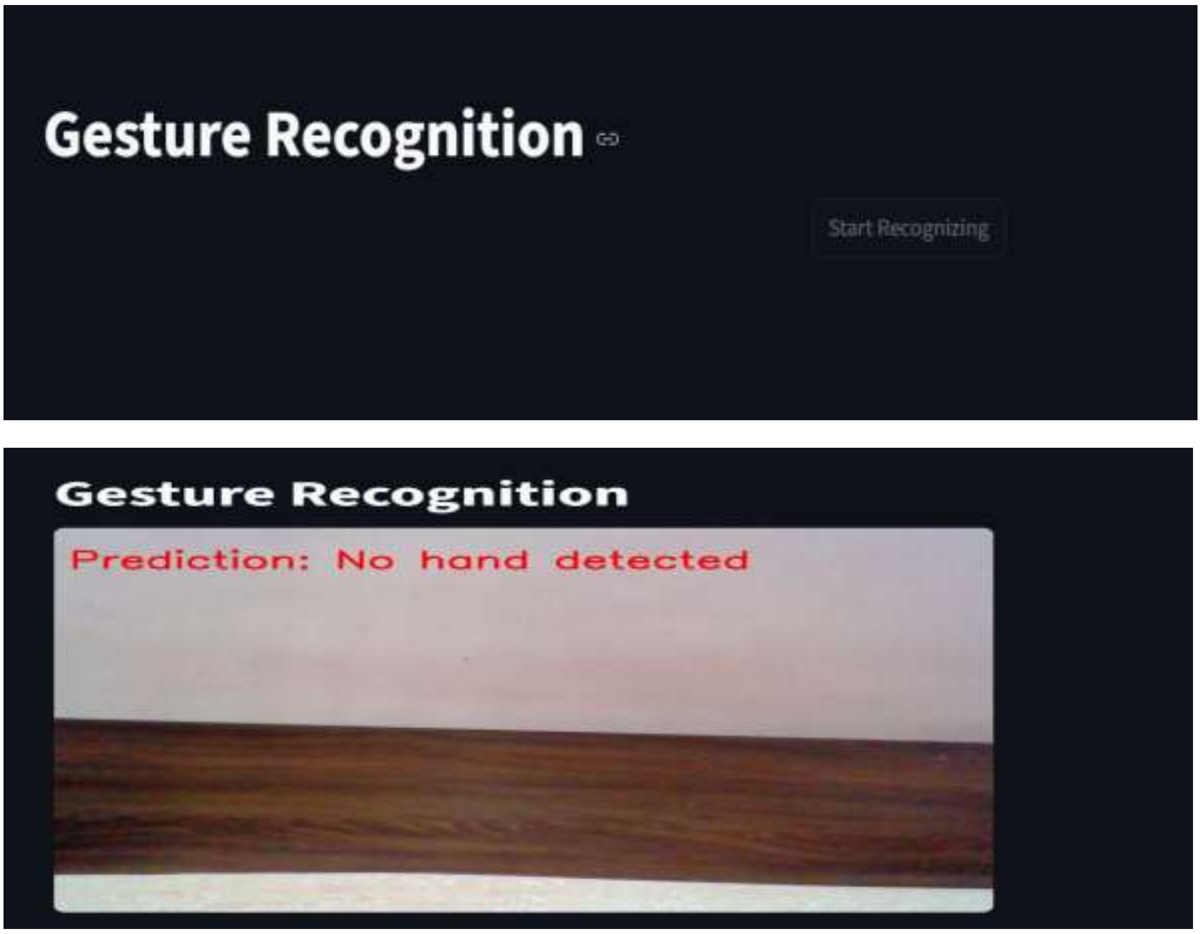
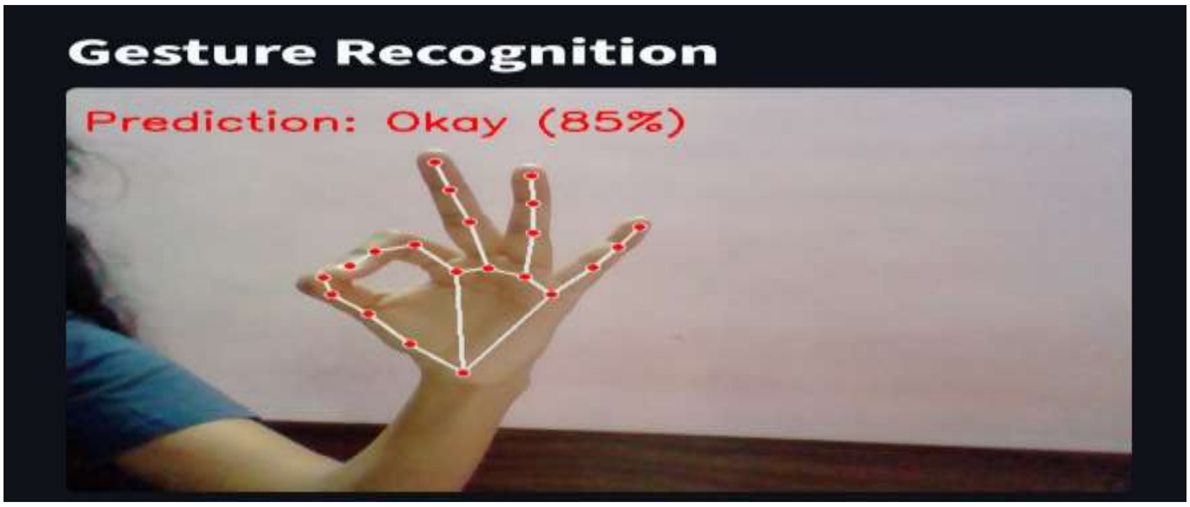


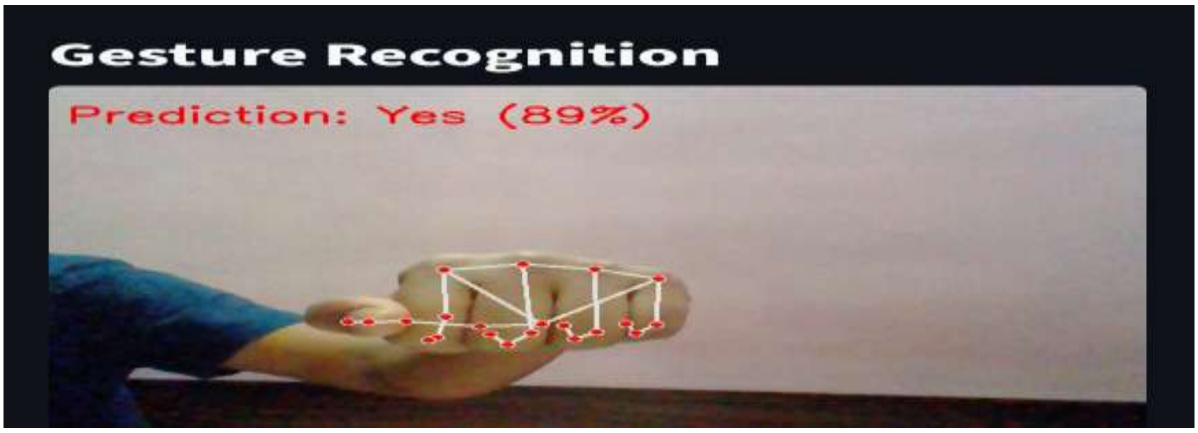
Fig. 6: Live sign language recognition interface

5.3 Prediction Output Display



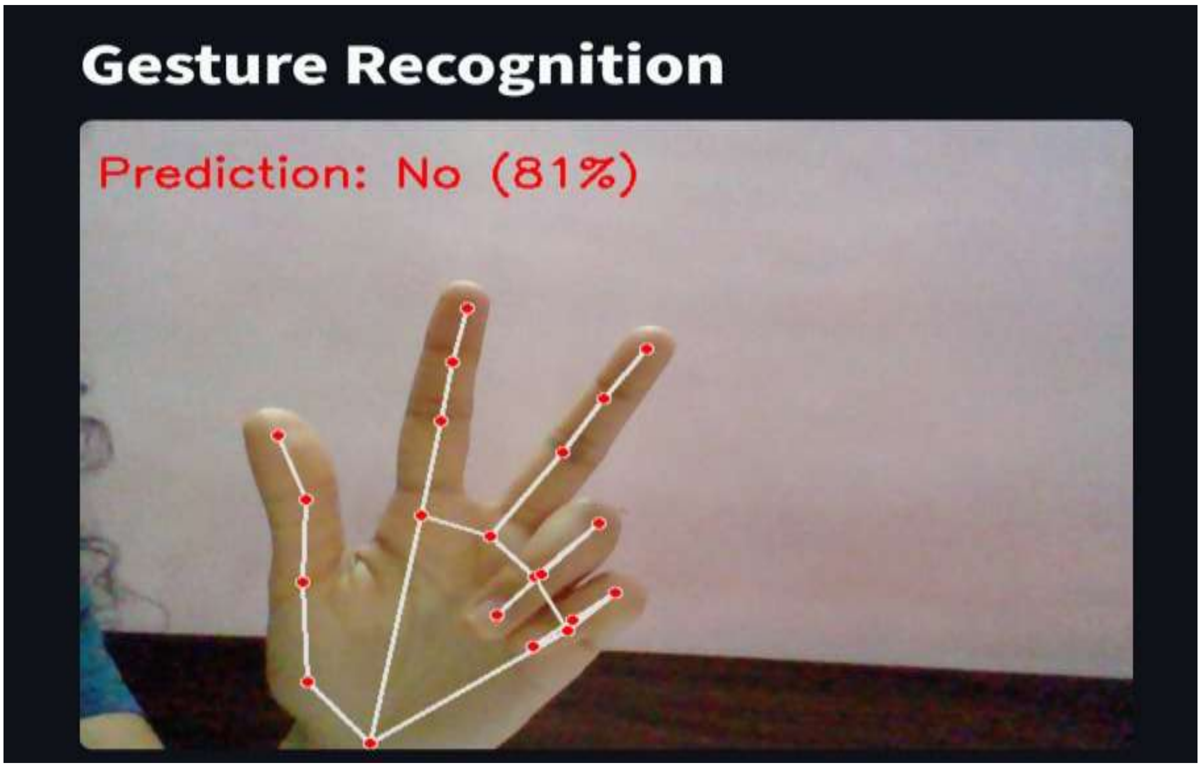
Gesture Recognition

Prediction: Okay (85%)



Gesture Recognition

Prediction: Yes (89%)



Gesture Recognition

Prediction: No (81%)

Fig. 7: Gesture prediction with confidence score

6. EXPERIMENTAL RESULTS AND DISCUSSION

The performance of SignSpeak was tested on multiple hand gesture samples. The gesture-wise accuracy is shown in the table below:

Gesture	Accuracy (%)
Hello	96
Yes	97
No	95
Okay	96
Please	97
Thank You	96

Real-time performance: The system runs at approximately 30 frames per second (FPS) on a standard laptop webcam.

Key observations:

- Normalizing the landmarks improves robustness against changes in hand distance and position [3].
- The lightweight Random Forest model allows fast predictions with low delay [4].
- The system performs very well for static gestures [4].
- Using a confidence threshold helps reduce incorrect predictions [4].
- The Streamlit dashboard allows users to visualize results in real time [4].

7. CONCLUSION AND FUTURE WORK

This study introduced SignSpeak, a real-time and user-friendly system for recognizing common sign language gestures. By leveraging MediaPipe Hands for hand landmark detection and a Random Forest classifier for gesture prediction, the system achieves high accuracy and operates smoothly without requiring powerful hardware.

Future work may include:

- Expanding the system to recognize a larger set of gestures and dynamic gesture sequences [4].
- Using temporal models such as LSTM to enable recognition of full sign language sentences [7].
- Deploying the system on mobile devices or cloud platforms to make it more accessible [6].
- Supporting multiple sign languages to increase usability across different regions [1].

8.REFERENCES

[1] A. K. Sahoo, G. S. Mishra, and K. K. Ravulakollu, “Sign Language Recognition: State of the Art,” *ARPN Journal of Engineering and Applied Sciences*, vol. 9, no. 2, Feb. 2014.

[2] D. Serai, I. Dokare, S. Salian, P. Ganorkar, and A. Suresh, “Proposed System for Sign Language Recognition,” in *Proc. ICCPEIC*, IEEE, 2017.

[3] S. Dessai and S. Naik, “Literature Review on Indian Sign Language Recognition System,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 9, no. 7, 2022.

[4] A. Pathak, A. Kumar, P. Priyam, P. Gupta, and G. Chugh, "Real-Time Sign Language Detection," *International Journal for Modern Trends in Science and Technology*, vol. 8, no. 1, pp. 32–37, 2022.

[5] S. Srivastava, A. Gangwar, R. Mishra, and S. Singh, "Sign Language Recognition System Using TensorFlow Object Detection API," in *Proc. Int. Conf. on Advanced Network Technologies and Intelligent Computing (ANTIC)*, 2021.

[6] M. Chitampalli, D. Takalkar, G. Pillai, P. Gaykar, and S. Khubchandani, "Real-Time Sign Language Detection," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 5, no. 4, pp. 2983–2986, 2023.

[7] M. H. Monisha, M. B. S. Manish, R. R. Iyer, and S. J. Siddarth, "Sign Language Detection and Classification Using Hand Tracking and Deep Learning in Real-Time," *International Research Journal of Engineering and Technology (IRJET)*, vol. 10, no. 11, pp. 875–881, 2023.

[8] S. Dutta, A. Bose, S. Dutta, and K. Roy, "Sign Language Detection Using Action Recognition with Python," *International Journal of Engineering Applied Sciences and Technology*, vol. 8, no. 1, pp. 61–67, 2023.

[9] R. K. Pathan *et al.*, "Sign Language Recognition Using the Fusion of Image and Hand Landmarks Through Multi-Headed Convolutional Neural Network," *Scientific Reports*, vol. 13, Art. no. 16975, 2023.