Real-Time Sign Language to Speech Converter Using MediaPipe and OpenCV

Dr. K N. Tayade[1], Aman Mahesh Mishra[2], Abhay Gajanan Chorey[3],
Vedang Anand Nimdeokar[4], Sanketsingh Jayantiprasad Thakur[5]
[1,2,3,4,5]*Department of Information Technology, GCOE, Amravati*

***Abstract*—The communication problem of specially enabled people is very crucial for their literacy and development. This paper presents a Real-Time Sign Language to Speech Converter using Computer Vision and Deep Learning. The system captures hand gestures through a camera and processes them using OpenCV and NumPy. A Keras-based model trained on a custom dataset recognizes gestures and converts them into text. The generated text is then converted into speech using a Text-to-Speech system. This paper presents the solution to bridge the communication gap between hearing-impaired individuals and others by enabling real-time and effective interaction.**

I. INTRODUCTION

Communication plays a vital role in human interaction, but individuals with hearing and speech impairments primarily rely on sign language, which is not widely understood by everyone. This creates a significant communication gap in daily life. To address this issue, this paper proposes a Real-Time Sign Language to Speech Converter using Computer Vision and Deep Learning techniques [3],[10]. The system captures hand gestures through a camera and processes them using image processing methods. A trained Keras model recognizes the gestures and converts them into text [9], which is further transformed into speech using a Text-to-Speech module [10]. This approach aims to enable smooth and real-time communication, making it easier for hearing-impaired individuals to interact with others.

II. LITERATURE SURVEY

Sign language recognition (SLR) has evolved from early sensor-based systems to modern vision-based approaches using computer vision and machine learning. Earlier methods relied on wearable devices like data gloves, which were accurate but impractical. Recent systems instead use cameras and frameworks such as MediaPipe to detect hand landmarks in real time [1]. By extracting 21 keypoints per hand, these systems create efficient feature representations that reduce computational cost while maintaining accuracy, making them suitable for real-time gesture recognition applications.

Traditional machine learning techniques, particularly the Random Forest classifier [5], have been widely used for static gesture recognition due to their robustness and ability to handle high-dimensional data. However, such models are limited when dealing with dynamic gestures that involve temporal variation. To address this, deep learning approaches like Long Short-Term Memory (LSTM) networks have been introduced, as they effectively capture sequential dependencies and motion patterns [11]. Convolutional Neural Networks (CNNs) have also been widely applied for feature extraction in gesture recognition tasks [3], [10], enabling recognition of alphabets and continuous gestures.

Recent research emphasizes hybrid systems that combine machine learning and deep learning for improved performance and efficiency. These systems often integrate tools like OpenCV for video processing [7] and frameworks such as TensorFlow and Keras for model development and deployment [8], [9]. Despite advancements, challenges such as variability in hand appearance, occlusion, and limited datasets persist. Ongoing work focuses on improving robustness and scalability, aiming to make SLR systems more accurate and practical for real-world communication support.

III. METHODOLOGY

The proposed sign language recognition system follows a structured pipeline consisting of data collection, feature extraction, model training, and real-time prediction. The dataset is organized into labeled folders representing different gestures such as YES, NO, HELLO, and STOP. Each folder contains multiple samples stored as NumPy arrays, where each file corresponds to a single captured gesture instance. This organized structure enables efficient data loading and labeling during the training phase. Figure Fig.1 shows the propsed architecture of the system.

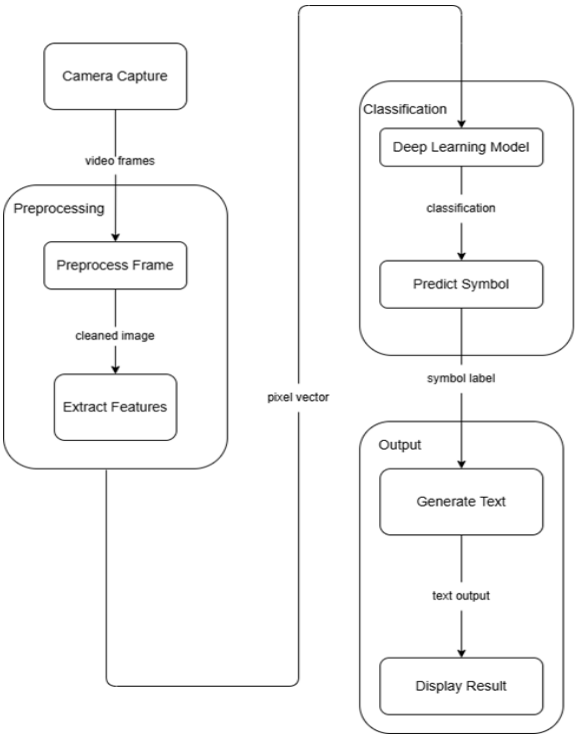

Fig.1. Architecture of the Sign Language Recognition System

For feature extraction, hand landmarks are obtained using MediaPipe, which detects 21 keypoints of the hand in three-dimensional space. Each landmark consists of x, y, and z coordinates, resulting in a total of 63 features per sample. To improve model robustness and eliminate positional dependency, all landmark coordinates are normalized by subtracting the wrist coordinates. This transformation ensures that the system focuses on the relative structure of the hand rather than its absolute position in the frame.

The classification stage employs a dual-model approach to handle different types of gestures. For static word-level gestures, a Random Forest classifier is used. The model is trained on the extracted feature vectors and leverages multiple decision trees to improve prediction accuracy and reduce overfitting. During inference, the model outputs class probabilities, and a prediction is accepted only if the confidence exceeds a predefined threshold, ensuring reliable recognition.

For alphabet recognition, a deep learning model based on Long Short-Term Memory is utilized. The extracted features are reshaped into a sequence format and passed to the LSTM model, which is capable of learning patterns in sequential data. The model produces probability scores for each alphabet class, and the class with the highest probability is selected as the predicted output.

The system is deployed as a real-time application using OpenCV for video capture and frame processing, and Streamlit for building an interactive user interface. Video frames are continuously captured from the webcam, processed to extract hand landmarks, and fed into the trained models for prediction. The predicted gesture, along with its confidence score, is displayed in real time.

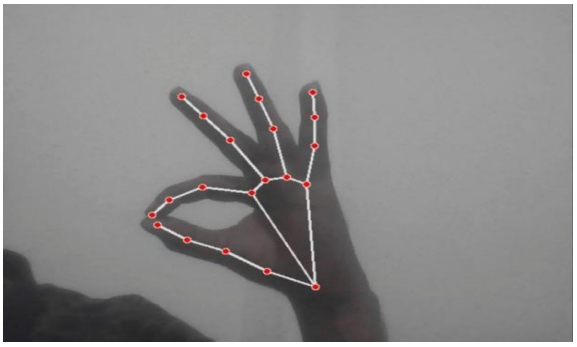

Fig. 2: Hand Landmark Detection using MediaPipe in Real-Time Sign Language Recognition System

Fig. 2 illustrates the prototype of the proposed Sign Language Recognition System during real-time execution. The system captures live video input through a webcam and processes it using MediaPipe to detect and track hand landmarks, as shown by the highlighted keypoints and connections on the hand. Each detected landmark represents a specific joint of the hand, forming a structured representation that is further converted into a feature vector. These features

are then passed to the trained model for gesture classification. The image demonstrates the system's ability to accurately detect hand pose in real time, even under varying lighting conditions, validating the effectiveness of the feature extraction and preprocessing stages in the proposed approach.

Additionally, the system maintains a dynamic sentence buffer where recognized words are appended sequentially. This allows continuous interpretation of gestures into meaningful text. The application supports two operational modes word mode and alphabet mode providing flexibility in usage while maintaining real-time performance and user-friendly interaction. Fig.3 shows the detailed data flow diagram of the system.

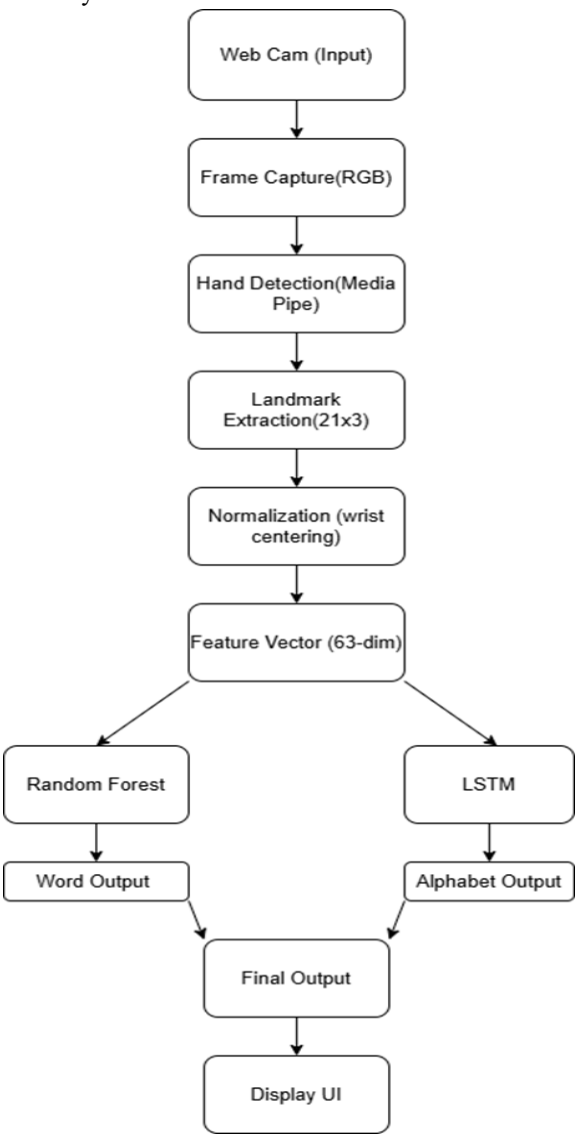

Fig.3.Data Flow Diagram of the Sign Language Recognition System

IV. RESULTS

The Sign Language Recognition System was successfully developed and evaluated to assess its effectiveness in recognizing hand gestures in real time. The system provides an interactive interface for capturing hand movements through a webcam and translating them into meaningful words and alphabets, significantly reducing the communication gap for users relying on sign language.

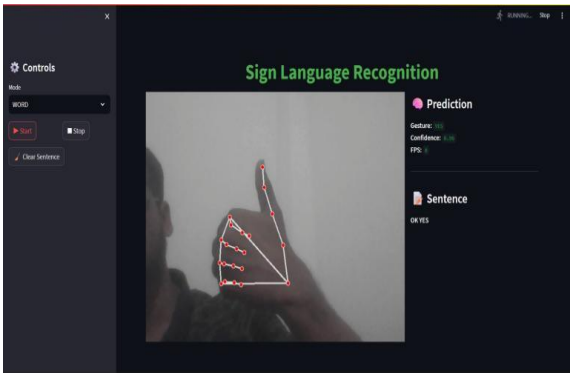

Fig. 4: Real-Time Sign Language Recognition and Output Display

The above figure presents the final output of the developed Sign Language Recognition System during real-time execution. The system effectively detects hand landmarks and processes them into structured feature representations, which are then used by the trained models to classify gestures accurately. The predicted gesture is displayed on the interface along with a confidence score and frame rate (FPS), providing immediate feedback to the user. In addition to individual predictions, the system maintains a dynamic sentence buffer, allowing consecutive gestures to be combined into meaningful text while avoiding repeated entries.

The results demonstrate that the system performs reliably under real-time conditions, with consistent prediction accuracy and smooth responsiveness. The use of normalized feature vectors ensures robustness against variations in hand position and orientation, while the hybrid modeling approach enables efficient handling of both static word-level gestures and sequential alphabet-based inputs. The system operates with low latency, ensuring a seamless user experience during continuous interaction.

Overall, the developed system achieves accurate gesture recognition, effective sentence formation, and stable real-time performance, indicating its suitability for practical applications in assistive communication for individuals with hearing or speech impairments.

V. CONCLUSION

The communication problem of specially enabled people can be solved by the Sign Language Recognition System. It demonstrates an effective approach for translating hand gestures into meaningful text using a combination of computer vision and machine learning techniques. By leveraging Media Pipe for accurate hand landmark extraction, the system is able to generate reliable feature representations that are robust to variations in hand position and orientation.

A hybrid classification strategy was employed to improve performance across different types of gestures. The use of a Random Forest model enabled efficient and accurate recognition of static word-level gestures, while the Long Short-Term Memory model facilitated the recognition of alphabet-based gestures by capturing underlying patterns in the data. This combination ensured both computational efficiency and flexibility in handling diverse gesture types.

The system was successfully implemented as a real-time application using OpenCV for video processing and Streamlit for user interaction. Features such as live gesture prediction, confidence scoring, and sentence formation contributed to an intuitive and responsive user experience.

REFERENCES

[1] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. Chang, and M. Grundmann, "MediaPipe Hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[2] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 10023–10033.

[3] [3] L. Pigou, S. Dieleman, P. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCV Workshops)*, 2014, pp. 572–578.

[4] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, 2015, pp. 1–8.

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[7] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[8] M. Abadi *et al*., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org

[9] F. Chollet, "Keras: Deep learning library for Python," 2015. [Online]. Available: https://keras.io

[10] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 3793–3802.

[11] Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin, Germany: Springer, 2012.