WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

HOLOSIGN-REALTIME SIGN LANGUAGE DETECTION SYSTEM

Amul S[1], Amruth B[2], Athersh JR[3], Ashwin Kumar B[4]

UG STUDENTS , SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY , COIMBATORE.

ABSTRACT :

In this paper, we propose a real-time system for recognizing sign language alphabets and basic commands using webcam-based hand landmark detection combined with machine learning. MediaPipe is utilized for precise extraction of 21 key hand landmarks in real time. These landmarks are converted into a normalized vector of coordinates, which serves as input for a Random Forest classifier. The system is capable of recognizing the 26 alphabets of the English language along with two additional signs for 'space' and 'delete'. We validate the model's performance through experiments conducted on a self-curated dataset. Our approach achieves over 92% classification accuracy and provides responsive real-time interaction, making it an effective solution for assistive communication.

Keywords: Sign Language Recognition, Hand Landmark Detection, Random Forest, MediaPipe, Assistive Communication, Real-Time Systems

1.INTRODUCTION

Millions of people worldwide rely on sign language for daily communication. However, the language barrier between sign language users and those unfamiliar with it continues to limit accessibility in many environments. Translators are not always available, and not everyone can learn sign language. Automated Sign Language Recognition (SLR) systems have gained attention as a potential solution to this problem. With advancements in computer vision and machine learning, SLR systems can now be implemented using consumer-grade cameras and efficient algorithms. In this work, we present a lightweight and fast sign language recognition system designed for real-time use. By combining hand tracking using MediaPipe and classification using a Random Forest model, our system efficiently converts hand gestures into corresponding textual characters. This aids seamless communication and can be extended to various assistive technologie.Deaf individuals frequently face difficulties in everyday communication due to the unavailability of interpreters or the lack of awareness of sign language. Innovations in artificial intelligence and computer vision present a unique opportunity to create intuitive systems that can bridge this gap. This project is driven by the aspiration to create inclusive technology, reduce communication disparities, and leverage machine learning for social good. Our system aims to be a real-time, non-intrusive, and user-friendly tool accessible to all.Despite the availability of advanced machine learning tools, real-time and affordable sign language recognition remains a challenge. Most existing systems are either limited in vocabulary, lack speed and accuracy, or rely on expensive and complex hardware. There is a strong need for a lightweight, accurate, and scalable solution that works efficiently using standard computing devices.

1.1 objectives

- To develop a system capable of recognizing all 26 ASL alphabet hand signs.
- To enable dynamic text formation using SPACE and DELETE functionality.
- To create a robust training dataset from scratch using MediaPipe hand landmarks.
- To train an accurate machine learning model using Random Forest.
- To deploy the model for real-time sign prediction using OpenCV.

1.2 scope of the project

The project is focused on American Sign Language (ASL) alphabets and operates in real time using a single webcam. While the current implementation covers alphabetic gestures and basic control signs, the architecture can be expanded in the future to include words, numbers, and contextual gestures. The system will be capable of functioning without internet access, making it ideal for offline and classroom scenarios

2. LITERATURE REVIEW

Over the years, several methodologies have been proposed for SLR. Early methods relied heavily on sensor gloves or motion capture systems to identify gestures. Although effective, these devices were cumbersome and cost-prohibitive. More recent efforts have turned to image-based recognition using

convolutional neural networks (CNNs). While powerful, CNNs often require large datasets, powerful hardware, and extended training time. Several studies have also explored the use of MediaPipe for hand tracking due to its real-time performance and robustness in dynamic backgrounds. The Random Forest algorithm has been employed in many gesture recognition tasks due to its speed, accuracy, and ease of implementation. Our approach draws on these previous works to offer a practical, scalable, and efficient solution for real-time alphabet-based SLR. Deep learning has revolutionized the field of image recognition, enabling computers to learn from raw data. Convolutional Neural Networks (CNNs) have been particularly effective in recognizing static hand gestures. Many studies have applied CNNs to classify ASL alphabet gestures with high accuracy. However, CNNs often require large training datasets and powerful hardware for effective training. These models also lack transparency in decision-making, which makes debugging and optimization hardHand landmark :detection significantly improves gesture recognition by focusing on key points of the hand rather than the entire image. Google's MediaPipe library is a popular framework that provides accurate 21-point hand landmark detection in real time. It enables compact and efficient feature extraction by removing irrelevant background information. Using these landmarks instead of raw images helps reduce computational cost and increases model generalization.

3. SYSTEM DESIGN

3.1 Hand Landmark Extraction with MediaPipe

We leverage the MediaPipe Hands solution by Google, which extracts 21 3D landmarks from live webcam input. These landmarks include critical points such as the tips and joints of each finger and the wrist. The coordinates (x, y, z) of these points form a high-dimensional feature vector (63 dimensions per frame). MediaPipe ensures robustness to variable lighting and backgrounds, making it highly effective in uncontrolled environments.

3.2 Feature Vector Normalization

To ensure the model generalizes across users with different hand sizes and camera positions, the feature vector is normalized relative to the wrist landmark. This transforms the absolute coordinates into a relative coordinate system, reducing dependency on the distance between the camera and the user. The normalization step is crucial in eliminating scale, translation, and rotation variations.

3.3 Random Forest Classification

For classification, we chose the Random Forest algorithm due to its interpretability, speed, and effectiveness with small to medium-sized datasets. Each sample from the training data consists of a 63-dimensional normalized feature vector and a label indicating the corresponding gesture. We use 100 trees in the forest with a maximum depth that prevents overfitting. The model is trained on 80% of the dataset and validated on the remaining 20%. Cross-validation techniques were used to ensure the model's robustness.

3.4 Real-Time Prediction Pipeline

Once the model is trained, it is integrated into a real-time Python application using OpenCV. The system reads webcam input, applies MediaPipe to detect landmarks, normalizes the extracted vector, and feeds it to the Random Forest model for classification. The predicted letter is appended to a string that forms a sentence, with special gestures recognized for 'space' and 'delete' commands. This allows users to construct full sentences intuitively and quickly.

3.5 Flow Diagram

4. EXPERIMENTAL SETUP AND DATASET

We built a custom dataset by recording video frames of users displaying each of the 26 alphabets and two additional commands. The recordings were made under various lighting conditions and backgrounds to improve the model's generalizability. Data was collected from five individuals with different hand shapes and skin tones. In total, over 5000 samples were gathered, balanced across the 28 categories. The data was manually labeled, and

augmentation techniques like rotation and scaling were applied to increase diversity. For training and testing, we split the dataset in an 80-20 ratio. The system was run on a standard laptop (Intel i5 CPU, 8GB RAM), confirming that the model can operate in real time without the need for high-end GPUs. We created a custom dataset with the following properties:

- **Classes**: 26 Alphabets + 2 symbols (SPACE, DELETE).
- **Images**: Around 300 images per class.
- **Conditions**: Different lighting, backgrounds, and slight variations in hand poses.
- **Capture Method**: Using OpenCV's VideoCapture module.

5.RESULTS AND EVALUATION

The trained Random Forest model achieved an average accuracy of 92.5% on the test set. The precision and recall for most classes were above 90%. Gestures like 'U', 'V', 'W', and 'Y' showed the highest classification rates, while confusion occurred between visually similar gestures like 'M' and 'N' or 'E' and 'F'. In real-time testing, the system maintained an average frame rate of 15–20 FPS, ensuring smooth user interaction. The latency from camera input to output display was less than 100 milliseconds. Comparative analysis shows that our method performs on par with CNN-based systems while requiring significantly fewer computational resources. The confusion matrix highlights specific areas for improvement in misclassified classes.

6. LIMITATIONS AND FUTURE ENCHANCEMENTS

While our system performs well in controlled conditions, several challenges remain. Motion blur caused by rapid hand movement can reduce landmark accuracy. The system may also struggle with multiple hands appearing in the frame or partially occluded gestures. Moreover, it currently only supports static alphabet gestures, excluding dynamic signs and full words. Future work includes expanding the vocabulary to include dynamic gestures and common sign words, incorporating NLP modules for grammar correction, and deploying the application on mobile devices or embedded systems such as Raspberry Pi. Using recurrent models or transformers could also help in recognizing sequences of gestures as words or sentences. Additionally, user personalization modules can be added to adapt the model to individual gesture styles.

7.CONCLUSION

In this work, we presented a practical and efficient system for real-time sign language recognition using hand landmark detection and machine learning. By employing MediaPipe and Random Forest, we achieved a balance between performance and computational efficiency. The system is capable of recognizing individual alphabets and basic commands, enabling users to form sentences on the fly. Our results validate the model's accuracy and responsiveness, demonstrating its potential for deployment in assistive communication tools for the hearing and speech-impaired community.

Future Enchancements:

- Deploy model on Android apps.
- Integrate with Augmented Reality (AR) devices.
- Use LSTM (Long Short-Term Memory) models to improve continuous gesture recognition.

8.REFERENCES

[1] Z. Zhang et al., "Hand Gesture Recognition Based on CNN," IEEE Access, vol. 6, pp. 7899–7906, 2018.
[2] F. Zhang and C. Xu, "MediaPipe Hands: On-device Real-time Hand Tracking," arXiv preprint arXiv:2006.10214, 2020.
[3] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[4] OpenCV Library, https://opencv.org/
[5] Scikit-learn, https://scikit-learn.org/ [6] MediaPipe by Google, https://mediapipe.dev/
[6] Mediapipe documentation – https://google.github.io/mediapipe
[7]Google,"MediaPipe Hands: On-device Real-time Hand Tracking," [Online]. Available: https://google.github.io/mediapipe/solutions/hands\
[8] D. Amrita and S. P. Maity, "Real-time sign language recognition using CNN," *International Journal of Computer Applications*, vol. 182, no. 32, pp. 20-25, 2019.
[9] S. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," *Advances in Neural Information Processing Systems*, 2014.
[10] G. Papandreou, T. Zhu, and K. Murphy, "PersonLab: Person pose estimation and instance segmentation," *Proceedings of IEEE CVPR*, 2018.