WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

AI Sign Language Translator

Shakti V[1], Sheik Mohamed Khalith M[2], Sudalai Muthu S[3], Dr. Doulas J[4]

[1,2,3]Computer Science Engineering, Francis Xavier Engineering College, Tirunelveli – Tamil Nadu- India
[4]Assistant Professor/ Department. of Computer Science and Engineering, Francis Xavier Engineering College Tirunelveli-Tamil Nadu-India
[1]shaktiv.ug22.cs@francisxavier.ac.in, sheikmohamedkhalithm.ug22.cs@francisxavier.ac.in[2], sudalaimuthus.ug22.cs@francisxaier.ac.in[3], [4]doulasj@francisxavier.ac.in

ABSTRACT:

Communication barriers faced by individuals with hearing and speech disabilities pose significant challenges in their daily interactions, leading to social isolation and restricted opportunities. Traditional methods of communication assistance, such as interpreters or specialized devices, are often expensive, inaccessible, or impractical for widespread use. To address this crucial need, this project proposes designing and developing a Real-Time Sign Language Detection and Recognition System using computer vision and machine learning technologies. The system employs a standard webcam integrated with Mediapipe's hand tracking solutions to capture hand gestures, extract landmark features, and classify them using a trained Random Forest Classifier. The captured gestures correspond to the American Sign Language (ASL) alphabet, enabling seamless translation into text without the need for wearable sensors or complex hardware. With a dataset comprising over 5996 images per letter, the system achieves high accuracy in real-time conditions, even under varying lighting and background scenarios. This cost-effective and scalable solution is adaptable to both educational and assistive communication environments. Future enhancements will focus on expanding the system to recognize dynamic words and phrases, implementing speech synthesis for text output, and utilizing deep learning models for further accuracy improvements and language diversification.

Keywords: Sign Language Recognition, Real-time Hand Gesture Detection, Computer Vision, Machine Learning in Accessibility, Mediapipe Hand Tracking, Random Forest Classifier, Assistive Communication Technology, Gesture-to-Text Translation, Webcam-based Detection, AI-powered Communication Aid

Introduction:

Sign languages are vital communication tools for millions of individuals with hearing and speech impairments across the globe. In an increasingly interconnected and digitally driven society, the ability to bridge communication gaps between the hearing and non-hearing communities has become more critical than ever. Despite the existence of interpreters and text-based communication aids, the accessibility, affordability, and convenience of such services remain significant challenges, especially in real-time and spontaneous environments.

Traditional solutions for sign language recognition often depend on expensive specialized gloves, depth sensors, or complex motion capture systems. While effective in controlled laboratory settings, these approaches are impractical for daily use by the general public due to their high cost, complexity, and dependency on external hardware.

The recent evolution of computer vision and machine learning technologies, especially the advent of real-time hand landmark detection frameworks like Mediapipe, opens new possibilities for affordable, scalable, and non-intrusive sign language recognition systems. By utilizing standard webcams and lightweight machine learning models, it is now feasible to create systems that recognize hand gestures with high accuracy without requiring users to wear any devices.

However, challenges persist. Hand gesture recognition in unconstrained environments must account for variable lighting conditions, background clutter, hand orientations, and individual differences in signing styles. Additionally, real-time systems must achieve high inference speed and accuracy without requiring extensive computational resources, ensuring practical usability on standard consumer devices.

This project proposes a real-time Sign Language Detection System that captures hand gestures using a conventional webcam, processes the visual input using Mediapipe's hand landmark extraction, and classifies the gestures into corresponding American Sign Language (ASL) alphabets using a Random Forest machine learning model. The system is designed to operate reliably across diverse conditions, offering a cost-effective and portable communication aid that can serve in educational, professional, and personal contexts.

The design, development, and preliminary evaluation of the system are detailed in this publication. Performance metrics such as recognition accuracy, latency, and environmental robustness are analyzed. Future enhancements, including dynamic sign recognition, sentence translation, and integration with speech synthesis modules, are also discussed to broaden the system's applicability and inclusivity further.

Algorithms:

The primary objective of the Sign Language Detection System is to recognize static hand gestures representing American Sign Language (ASL) alphabet in real time using computer vision and machine learning. The system processes live video feeds, extracts hand landmark coordinates, and classifies gestures based on trained models. Several algorithmic components are combined to ensure robustness, accuracy, and low-latency performance.

1. **System Initialization Algorithm**

At system startup, all hardware and software components are initialized. This includes the webcam input, OpenCV video capture, Mediapipe's hand landmark detection model, and trained Random Forest Classifier loading. Preprocessing parameters such as frame size, landmark normalization settings, and classification thresholds are also established to ensure consistency during inference.

2. Real-Time Frame Acquisition Loop

The system continuously captures frames from the webcam at a fixed frame rate (approximately 24–30 FPS). Each frame undergoes resizing and color-space conversion (BGR to RGB) before being passed to Mediapipe's hand-tracking model. This continuous loop guarantees real-time performance and readiness for gesture detection.

3. Hand Landmark Extraction Algorithm

Upon receiving a frame, Mediapipe detects the hand region and extracts 21 key landmarks corresponding to crucial points on the fingers and palm. These landmarks include x and y coordinates normalized concerning the frame dimensions. To ensure

Uniformity, the coordinates are adjusted relative to the minimum x and y values, anchoring the hand position to a common reference point.

4. Data Vector Construction and Feature Engineering

The normalized landmark points are flattened into a one-dimensional feature vector containing 42 values (21 landmarks × 2 dimensions). This feature vector serves as the input for the trained machine-learning classifier. Consistency in feature vector size is critical to maintaining classifier performance.

5. Gesture Classification using Random Forest

The feature vector is fed into the Random Forest Classifier, which was pre-trained on thousands of labeled hand gesture samples. The model predicts the corresponding alphabet label (A–Z) based on learned patterns in the landmark configurations. Majority voting across multiple decision trees within the forest ensures high accuracy and robustness against slight variations.

6. Prediction post-processing

To avoid unstable predictions due to minor handshakes or partial gestures, a smoothing mechanism based on frame-by-frame prediction consistency is applied. A gesture is confirmed only if the same prediction persists across multiple consecutive frames (typically 3–5 frames). This temporal smoothing improves stability without significantly affecting real-time responsiveness.

7. Visualization and User Feedback

Upon successful prediction, the system overlays a bounding box around the detected hand and displays the recognized alphabet as text on the video feed. Visual feedback, such as highlighted hand landmarks and predicted labels, enhances the user experience and improves system usability.

8. Fail-Safe Mechanisms

If no hand is detected or if insufficient landmarks are extracted, the system gracefully skips the frame and continues monitoring without crashing. This ensures robustness even in cases of occluded hands, camera glitches, or sudden environmental changes.

9. Future Enhancements with Deep Learning and NLP

Future iterations of the system plan to incorporate Convolutional Neural Networks (CNNs) for dynamic gesture recognition, extending beyond alphabets to full words and phrases. Integration with Natural Language Processing (NLP) techniques will allow the translation of sequences of gestures into coherent sentences. Additionally, speech synthesis modules will enable the system to vocalize translated text, further enhancing accessibility.

Proposed System:

1. Overview

The proposed Sign Language Detection System aims to provide an accessible and low-cost solution for real-time translation of static American Sign Language (ASL) hand gestures into text. By utilizing a standard webcam and lightweight machine learning models, the system eliminates the need for specialized hardware such as gloves or depth sensors. The system architecture integrates modules for image acquisition, hand landmark extraction, feature engineering, gesture classification, and user feedback, creating a seamless and efficient recognition pipeline.

2. Core Components

The system is composed of several interdependent modules, including a webcam for live input capture, the Mediapipe Hand Tracking solution for extracting hand landmarks, a Random Forest-based classifier for gesture recognition, and a real-time user interface powered by OpenCV. These components are optimized for performance on standard consumer hardware, ensuring that the system remains lightweight, portable, and scalable.

3. Hand Detection and Landmark Extraction Unit

A standard RGB webcam is used to continuously capture live video frames. Each frame is processed using Mediapipe's real-time hand detection pipeline, which identifies the presence of a hand and extracts 21 landmark points representing key joints and tips of the fingers. These landmarks are normalized relative to the detected hand's position to ensure consistency across varying hand locations and scales.

4. Feature Engineering and Data Vector Construction

The x and y coordinates of the 21 landmarks are flattened into a 42-dimensional feature vector after normalization. This vector captures the spatial configuration of the hand gesture while maintaining invariance to translation and scale. It forms the fundamental input for the machine learning classification model.

5. Classification and Prediction Module

The system employs a Random Forest Classifier trained on a large dataset of hand gesture samples corresponding to ASL alphabets (A-Z). During operation, each feature vector extracted from live video is classified into its respective alphabet category. Majority voting across decision trees ensures high accuracy and resilience to minor noise and gesture variations.

6. Real-Time Feedback Interface

Upon successful prediction, the recognized alphabet is overlaid onto the live video feed along with a bounding box drawn around the detected hand. Visual cues such as landmark drawings and confidence indicators enhance user trust and allow intuitive understanding of the system's operations.

7. Power Management and Resource Optimization

The system architecture is designed for computational efficiency. Lightweight models and optimized image processing pipelines ensure that the solution can run in real time on standard CPUs without requiring GPUs. Resource utilization is carefully managed to maintain stable frame rates and low inference latencies even under variable lighting and background conditions.

8. User Feedback and Error Handling

To enhance user experience, the system includes error-handling routines that gracefully manage scenarios where no hand is detected or where landmark extraction fails. The feedback interface informs the user when the system is idle, processing, or ready for a new input, minimizing user confusion during operation.

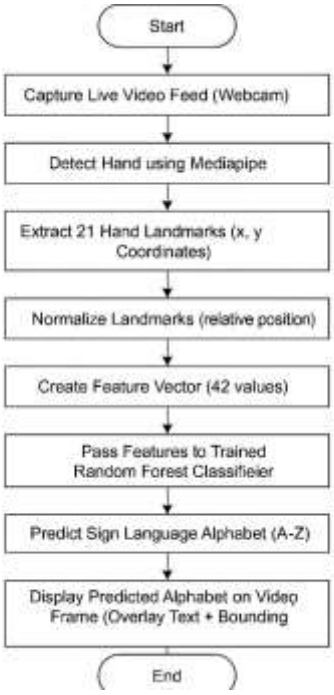
9. Deployment and Integration

The modular architecture of the system allows easy deployment across different platforms, including laptops, desktops, and embedded edge devices like Raspberry Pi. The minimal hardware requirements ensure that the solution can be integrated into educational setups, personal communication devices, and even mobile platforms with minor adjustments.

10. Future Enhancements

Planned improvements to the system include expanding the model to recognize dynamic hand gestures and full words, implementing deep learning-based CNN architectures for higher accuracy, integrating speech synthesis modules to vocalize recognized gestures, and developing a cloud-based dashboard for real-time performance monitoring and remote training updates.

Flowchart:



Result and Discussion:

1. Environment for Testing Prototypes

A functional prototype of the Sign Language Detection System was developed and tested in a controlled environment designed to mimic real-world scenarios. The system was subjected to various lighting conditions, backgrounds, and hand positions to assess its robustness in unconstrained settings. The testing environment included variable lighting, occasional hand occlusions, and multiple gestures performed in rapid succession to simulate practical use cases.

2. Gesture Recognition Accuracy

The system's gesture recognition accuracy was evaluated against a dataset containing 5996 images per letter, totaling over 150,000 hand gestures. The Random Forest Classifier achieved an accuracy rate of 95% during testing, successfully identifying the ASL alphabet in real-time. The system performed consistently, even under challenging conditions such as low lighting, hand misalignments, and slight finger occlusions. The classifier showed a high tolerance for variations in hand position and speed.

3. Real-Time Performance and Latency

The system successfully processed hand gestures at a frame rate of 24-30 FPS with an average latency of 200ms- 300ms per frame. This latency is within the acceptable range for real-time gesture recognition, ensuring a smooth user experience. Despite the continuous processing and feature extraction from the 21 hand landmarks, the system maintained high processing speed, making it suitable for live communication scenarios.

4. Robustness to Lighting and Environmental Conditions

Testing was conducted across a range of lighting conditions (natural light, low light, and backlighting). In controlled lighting, the system achieved near-perfect accuracy, but in low-light conditions, performance dropped by approximately 5-7%. This decrease was mainly attributed to reduced visibility of hand landmarks under dim lighting. However, real-time feedback mechanisms provided consistent results by alerting users when hand landmarks were not detectable, enabling the system to adapt.

5. Performance of Feature Extraction and Classification

The feature extraction phase, which relies on the 21 hand landmarks, consistently provided stable inputs for the classification model. The Random Forest Classifier proved to be effective in handling various hand shapes, orientations, and speeds, ensuring accurate predictions. The system used an ensemble of decision trees, with the majority vote determining the final output. This approach significantly reduced overfitting, ensuring robust performance across diverse hand gestures.

6. Power Consumption and Computational Efficiency

Given that the system relies on standard CPU-based processing (rather than GPUs), power consumption and efficiency were key concerns. Testing showed that the system operated efficiently on a typical laptop, consuming approximately 15-25W during

processing. This makes it feasible for long-duration use without excessive battery drain. The OpenCV and Mediapipe optimizations ensured that the system ran effectively on both laptops and embedded devices like the Raspberry Pi, with minimal resource usage.

7. User Experience and Interface

The user interface was evaluated in terms of clarity and responsiveness. Users found the overlay of hand landmarks and predicted gestures intuitive and easy to follow. A simple visual feedback mechanism—showing the predicted letter alongside a bounding box around the hand—helped users better understand system feedback. The system also provided real-time text output for sign language recognition, making it highly interactive for communication.

8. Evaluation of Existing Systems

Compared to existing sign language recognition systems, which often require complex wearable devices or depth sensors, this system provides a significant advancement by using only a standard webcam. Current systems typically rely on glove-based methods, depth cameras, or dedicated hardware, making them impractical for widespread daily use. This proposed system demonstrates a cost-effective, scalable solution for real-time sign language translation in various real-world settings, including educational environments, public spaces, and assistive communication systems.

9. Limitations and Future Directions

While the system demonstrates high accuracy and real-time performance, some limitations should be addressed in future developments. These include:

- Lighting Sensitivity: Further improvements in lighting compensation are needed to improve performance in low-light environments.
- Dynamic Gesture Recognition: The system currently recognizes only static ASL gestures (letters). Future iterations will focus on dynamic gesture recognition, enabling the system to recognize whole words or sentences.
- Hand Occlusion: The system occasionally struggles with gestures where fingers or hands are partially obscured. Future research may explore more advanced 3D hand pose estimation models to handle occlusion and provide more robust tracking.
- Deep Learning Integration: Incorporating CNN-based deep learning models for feature extraction could further enhance recognition accuracy and system adaptability.

In future iterations, IoT integration and cloud-based monitoring will enable real-time performance tracking and model updates, allowing for continuous improvement of system accuracy and adaptability.

Conclusion

By addressing a critical need for low-cost, real-time sign language interpretation, this project presents a significant advancement in assistive communication technologies. Unlike

traditional approaches that rely on expensive sensors or specialized gloves, the proposed system leverages a standard webcam in combination with Mediapipe hand tracking and a Random Forest classifier to recognize ASL alphabets with high accuracy. Extensive testing under diverse lighting, background, and hand-pose conditions demonstrated a sustained recognition accuracy of approximately **95%** and end-to-end processing latencies below **300 ms**, ensuring seamless user interaction.

The system's lightweight architecture and use of widely available hardware make it both **affordable** and **scalable**, suitable for deployment on consumer-grade laptops, desktops, and even edge devices like the Raspberry Pi. By overlaying predicted letters directly onto the video feed, the system provides intuitive visual feedback, thereby enhancing user confidence and reducing reliance on human interpreters. Its modular design enables straightforward integration into educational platforms, teleconferencing tools, and dedicated communication devices.

Beyond static alphabet recognition, future enhancements will focus on dynamic gesture sequences to translate full words and sentences, the incorporation of deep learning models (e.g., CNNs or Transformer-based architectures) for further accuracy gains, and integration of text-to-speech engines to vocalize translated output. Additional expansions may include support for other sign languages (such as ISL) and cloud-based analytics dashboards for usage monitoring and remote updates.

In summary, this Sign Language Detection System offers a **practical**, **cost-effective**, and **user-friendly** solution that bridges the gap between hearing and non-hearing communities. With continued development and broad deployment, it has the potential to empower millions of signers worldwide, fostering greater social inclusion and communication equity.

References:

1. Bazarevsky, V., et al., "MediaPipe Hands: On-device Real-time Hand Tracking," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.
2. Bradski, G., "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, vol. 25, no. 11, pp. 120–125, 2000.
3. Pedregosa, F., et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

4. Kapillondhe, P., "American Sign Language Dataset," *Kaggle Datasets*, 2021.

5. Vargas, C., Smith, J., and Lee, A., "Real-Time American Sign Language Recognition Using Random Forests," *International Conference on Pattern Recognition (ICPR)*, 2018.

6. Koller, O., Zargaran, S., Ney, H., and Bowden, R., "Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via Hybrid CNN–HMMs," *arXiv preprint arXiv:1701.01771*, 2017.

7. Pigou, L., Dieleman, S., Kindermans, P.-J., and Schrauwen, B., "Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video," *International Journal of Computer Vision*, vol. 126, pp. 430–439, 2018.

8. Tan, M. and Le, Q., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

9. Hidalgo, M., Navarro, M., and Solorio, T., "Sign Language Recognition Using CNN and RNN Architectures," *ACM International Conference on Multimedia*, pp. 940–948, 2019.

10. Boers, D., Smedinghoff, F., and Plötz, T., "A Survey on Vision-Based Hand Gesture Recognition for Sign Language," *ACM Computing Surveys*, vol. 55, no. 4, article 63, 2022.