AI-ASL: Advancing Accessibility Through Intelligent Sign Language Recognition

Bineeth Reddy Narra
School of Computing and Engineering
Quinnipiac University
Hamden, Connecticut
BineethReddy.Narra@qu.edu

Chetan Jaiswal
School of Computing and Engineering
Quinnipiac University
Hamden, Connecticut
Chetan.Jaiswal@qu.edu

Kruti Shah
School of Computing and Engineering
Quinnipiac University
Hamden, Connecticut
Kruti.shah@qu.edu

Jonathan Blake
School of Computing and Engineering
Quinnipiac University
Hamden, Connecticut
Jonathan.Blake@qu.edu

Brian O'Neill
School of Computing and Engineering
Quinnipiac University
Hamden, Connecticut
Brian.ONeill@qu.edu

Abstract- American Sign Language (ASL) is a vital means of communication for the Deaf and Hard of Hearing (DHH) community. However, non-ASL users often struggle to understand it, leading to communication barriers. This research introduces a real-time ASL recognition system capable of detecting Single hand static gestures made with right hand, including 20 letter signs, and 16 static sign gestures, and all digits, as well as four accurately trained two hand static gestures. The system is structured in two layers: Layer 1 leverages Mediapipe for landmark-based tracking of single-hand gestures, while Layer 2 employs a Convolutional Neural Network (CNN) to classify two-hand static gestures, addressing challenges related to overlapping hands and complex gesture interactions. To enhance accuracy, a green screen-based preprocessing method is implemented, ensuring clean hand segmentation without background interference. Additionally, a foot pedal switch is introduced for gesture confirmation, reducing human bias and accidental misclassifications. The proposed system is lightweight, non-intrusive, and optimized for real-time execution, making it highly suitable for assistive communication, educational environments, and accessibility applications. Through this approach, we aim to develop a scalable and efficient ASL recognition system that fosters seamless interaction between ASL users and non-signers in daily life.

Keywords: American Sign Language (ASL), Convolutional Neural Network (CNN), Graphics Processing Unit (GPU), Central Processing Unit (CPU) Artificial Intelligence (AI), Computer Vision

I. INTRODUCTION

Communication plays a vital role in human interaction, but for the Deaf and Hard of Hearing (DHH) community, expressing themselves to non-American Sign Language (ASL) users remains a significant challenge. Bridging this gap is crucial for improving accessibility, and ensuring equal participation in educational, professional, and social settings. While interpreters and closed captioning services exist, they are not always readily available, leading to communication barriers that affect the daily lives of ASL users. A robust, real-time ASL recognition system can significantly enhance accessibility and independence for the DHH community, making interactions with non-signers smoother and more natural.

We present a real-time ASL recognition system that integrates MediaPipe [1] for single-hand gesture tracking and a Convolutional Neural Network (CNN) - based model for two-hand static gesture classification. Our system currently supports 20 alphabet signs, and 16 static signs, and digits (0 - 9) for single-hand gestures made with right hand, along with four accurately trained two-hand gestures.

Initially, we explored a rule-based approach using hand landmark distances and angles for classification. While effective for simple static gestures, this method struggled with two-hand gestures due to increased complexity and overlapping landmarks. It also showed limitations in similar-looking signs, leading to frequent misclassifications. To overcome these challenges, we transitioned to a CNN-based model for two-hand recognition, which improved accuracy and adaptability under real-world conditions.

To enhance recognition reliability, we introduced a green screen preprocessing technique for cleaner hand segmentation and a foot pedal switch for intentional gesture confirmation. These additions reduce noise from background interference and eliminate accidental misclassifications caused by involuntary hand movements.

This research aims to develop a fast, accessible, and practical ASL recognition tool that bridges the communication gap between ASL users and non-signers. Our hybrid, real-time system offers improved usability across ASL contexts while remaining lightweight.

II. RELATED WORK

The field of ASL recognition has seen substantial progress with various approaches aimed at interpreting static and dynamic gestures. In early stages of this field, many systems were based on color segmentation and feature-based techniques, such as HSV and YCbCr color models, to detect hand regions. For instance, researchers proposed using HSV segmentation for recognizing static gestures with relatively high accuracy, although these approaches were highly sensitive to lighting and background noise [2].Inspired by these limitations, we focused on robust preprocessing techniques like green screen and background removal in our own system for layer 2.

As deep learning gained traction, Convolutional Neural Networks (CNNs) were adopted [3] to improve the recognition accuracy of ASL gestures. (Ayush Agarwal) [4]demonstrated that CNN, Residual Network 50(ResNet50) are significantly enhancing recognition precision for static signs. Their implementation, while accurate, was still limited to single-hand gestures. This encouraged us to extend our work to multi-hand static gestures using CNN-based models in Layer 2 of our architecture.

Researchers have also explored MediaPipe for its lightweight, real-time hand-tracking capabilities. The article [5] on gesture recognition via landmark detection confirmed its effectiveness for identifying 21 key landmarks on the hand in real-time. These insights helped us adopt MediaPipe for our Layer 1, focusing on fast, single-hand gesture classification without requiring extensive training data. The paper [6] proposed a system using deep learning for sign language-to-alphabet conversion, so we extended our work with sign language detection to text and speech.

Finally, in terms of gesture recognition design, by focusing on preprocessing techniques like background subtraction and segmentation to enhance model performance. However, paper such as the one from [2] pointed out that while preprocessing with HSV and filters helped improve recognition. Based on this, we opted for green screen preprocessing for uniform segmentation, only after evaluating other background removal methods in our experiments.

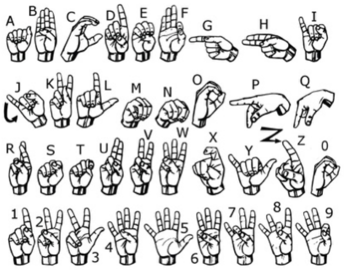


Fig 1. ASL Letters and Digits [7]

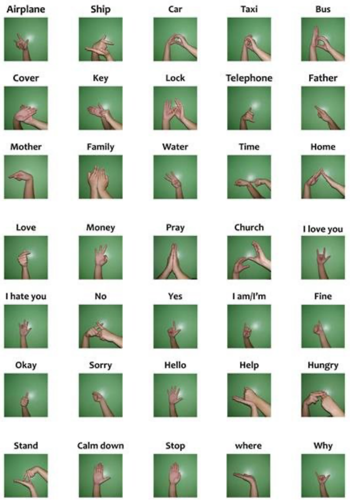


Fig 2. Static Word Signs [8]

From Fig. 1 [7] and Fig. 2 [8] , we are referencing the gestures for ASL letters, digits, and single-hand static signs. These images are used as the standard gestures that we followed during the development of our project. Throughout the project, our main focus was to detect these specific gestures accurately. We consistently referred back to these standard images to guide our implementation and testing, ensuring that the system we built could recognize and classify these gestures properly.

III. METHODOLOGY

This research presents a multi-layered ASL recognition system designed to efficiently detect and classify both single-hand and two-hand gestures in real time. The system integrates MediaPipe for landmark-based detection in Layer 1 and a CNN-based deep learning model in Layer 2, addressing challenges related to occlusions, overlapping hands, and complex gesture recognition. Additionally, preprocessing techniques such as green screen, segmentation ensure higher accuracy and robust performance across different environments.

However, gesture recognition is inherently challenging due to both human and machine limitations. One of the primary issues is the confusion between similar gestures, where certain ASL signs closely resemble others in terms of hand orientation and positioning. This results in misclassification, particularly in rule-based systems.

Another problem arises from gesture-to-gesture transitions, where computer vision models struggle to differentiate when one sign ends and another begins. Traditional gesture confirmation techniques, such as eye winks, head tilts, or double gestures, have been used to signal intentional inputs, but these methods introduce significant human bias. For example, some users may naturally blink more frequently than others, leading to unintended gesture confirmations. Additionally, head tilts or facial gestures require an additional layer of tracking, adding complexity to the recognition process and increasing computational overhead. This also makes it difficult for the system to determine whether the user has completed a sign or is still in the process of transitioning.

To address these limitations, this research incorporates a foot pedal switch for gesture confirmation, as shown in fig. 3. The foot pedal was chosen for its practicality and effectiveness in eliminating false positives. Unlike gesture-based confirmations such as head tilts or eye blinks which can occur unintentionally, the foot pedal requires a deliberate and conscious press by the user. This explicit action acts as a reliable confirmation signal, ensuring that only intended gestures are registered by the system. Additionally, because it does not rely on additional visual or motion sensors, this method simplifies the setup, keeping the system lightweight and computationally efficient.

Fig 3. Foot Pedal Switch

A. System Overview

1) Layer1-MediaPipe (Single-Hand Gesture Recognition): The first layer utilizes MediaPipe's hand-tracking model, which detects 21 key hand landmarks in real time, including fingertips, joints, and the palm base (as illustrated in Fig. 4). These landmarks are used to extract gesture features such as finger positions, angles, and distances, enabling efficient gesture classification without requiring deep learning-based processing. By using non-neural network methods, Layer 1 achieves real-time and computationally efficient classification of static single-hand gestures.

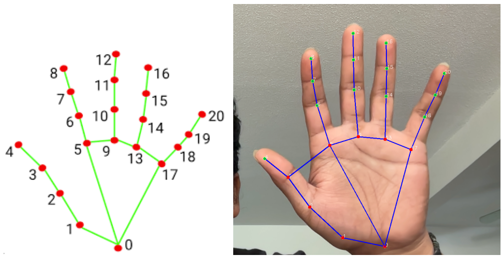


Fig. 4. Hand Landmarks Using Mediapipe [1]

This approach provides advantages like, firstly it eliminates the need for large-scale model training for single hand gestures, which makes the system lightweight and computationally efficient, and it maintains real-time processing speed using software libraries cvzone and mediapipe by ensuring that gesture recognition occurs instantaneously. Finally, because it relies on feature-based detection rather than color models, it performs reliably under varied lighting conditions, enhancing its robustness in diverse environments.

We implemented a Point-in-Polygon (PIP) algorithm (refer to fig. 4) to determine finger states open, half-closed, or fully closed based on the relative positions of the fingertips to the palm. This method enables accurate detection of static ASL signs by categorizing the degree of finger openness. The system uses 21 unique landmarks to track finger positions and the overall palm structure. In our approach, red-dotted region forms a polygon that helps classify gestures through distance-based analysis by considering if any fingertip comes under this polygon region making the specific finger is closed and by distance calculation, we are checking the finger is half closed are open. To further enhance gesture interpretation, we calculate the angular displacement of the hand using landmark points [9]. Hand orientation is determined using a coordinate-based approach, referencing key joints relative to the palm base (landmark 0) and the middle finger knuckle (landmark 9), as illustrated in Fig. 5. By calculating the angle at which the hand is facing, we classify the hand orientation into specific directions, as outlined in Table 1.

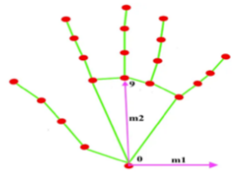


Fig. 5. Hand Tilt Calculation [9]

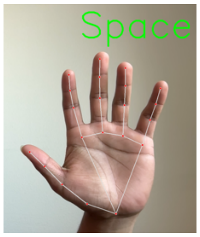


Fig 6. "*Space*" Gesture in Layer1

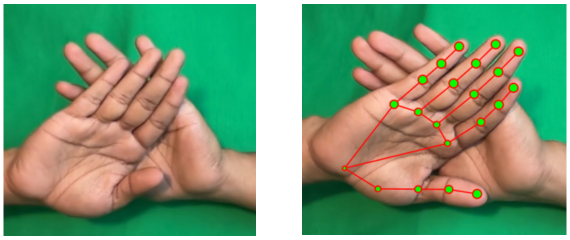
TABLE I: HAND ORIENTATION ANGLE THRESHOLD

Orientation	Angle Threshold
Upward	0° to 45° (m2 pointing up)
Downward	135° to 180° (m2 pointing down)
Left	45° to 135° (m2 pointing left)
Right	-45° to -135° (m2 pointing right)
Straight	Close to 90° (m2 parallel to x-axis)

To ensure smooth and accurate word formation, the system allows users to sequentially input letters through hand gestures. When a single-hand letter gesture is detected, it is temporarily stored in a list by clicking the foot pedal once. Additionally, a predefined space gesture (refer to Fig. 6) enables separation between words. When the user performs a double foot click, all collected letters are combined into a word and displayed on the screen. If spaces exist between words, the final phrase is synthesized into speech using Google Text-to-Speech enabling seamless communication between ASL and non-ASL users.

Despite these advantages, Layer 1 cannot effectively recognize complex two-hand gestures, as landmark-based distance calculations become inaccurate when hands overlap (see in fig 7.b). To overcome this, we implement Layer 2 for more advanced gesture classification.

2) Layer 2 – CNN-Based Two hand Gesture Recognition: Recognizing two hand gestures presents a greater challenge than single-hand detection due to overlapping hands, occlusions, and complex interactions that require deeper spatial and contextual understanding. Traditional landmark-based methods struggle with these issues as they primarily rely on fixed distance calculations between key points, which become unreliable when multiple hands are involved. To overcome these limitations (See Fig. 7.b), we implemented a Convolutional Neural Network (CNN) in Layer 2 to improve classification accuracy for two hand gestures.



a. Two Hand Gesture ("*Cover*") b. Layer-1 inept to detect 2 hands

Fig 7. Two-Hand Static Gesture (*"Cover"*)

CNNs were chosen for this task because they excel at feature extraction and pattern recognition in images, making them well-suited for analyzing hand structures, orientations, and inter-hand relationships. Unlike rule-based approaches, CNNs automatically learn spatial hierarchies of features, reducing the need for manual feature engineering.

We trained our model on a dataset comprising four specific two-hand ASL gestures (in fig 11.a second row). To improve recognition accuracy and ensure the model could differentiate between actual signs and random hand configurations, we expanded the dataset with five additional classes: right hand, left hand, two hands, back hands, and no hands. This allowed the model to first detect hand presence and orientation before classifying the actual gesture, making the system more robust and adaptable to varied hand positions and real-world scenarios. The green screen preprocessing technique (illustrated in Fig. 11.a.) ensures that the model isolates hand regions effectively, removing background noise and improving classification performance.

Furthermore, CNNs provide robust adaptability, allowing the model to recognize dynamic hand movements even under different lighting conditions or backgrounds. To enhance accuracy, we applied preprocessing techniques such as green screen background removal, ensuring that the model focuses solely on the hand features without distractions.

Unlike Layer 1, which uses predefined rules for classification, Layer 2 automatically learns gesture patterns through training, enabling it to recognize complex two-hand interactions with higher accuracy. This learning-based approach helps reduce misclassifications caused by overlapping fingers and hand occlusions. The system switches from Layer 1 to Layer 2 (as shown in Fig 8.) with user's preference as user is given option to select the type of gestures he makes like digits, ASL letters and statics signs come under layer 1. User needs to select the option manually, later the gestures is detected without making any confusion with another section and in two hands there is no specific parts, so model able to detect the trained gestures. This hybrid architecture ensures that simple gestures are processed efficiently using lightweight rule-based methods, while more complex cases are handled using deep learning for better accuracy and robustness.

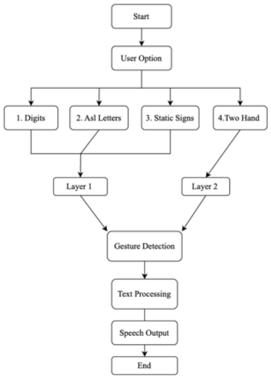


Fig 8.System Architecture

Step 1: Raw Color Images for Training

Initially, the model was trained using raw color images as shown in fig 9, capturing hand gestures in various lighting conditions and backgrounds. While this approach provided the advantage of working with unaltered, real-world data, it posed significant challenges. Background noise, varying lighting conditions, and distractions in the image often resulted in poor model performance. The model struggled to differentiate between hands and surrounding objects, leading to false positives and incorrect classifications. Although this step established a baseline for training, it was not sufficient for achieving reliable real-time gesture recognition.

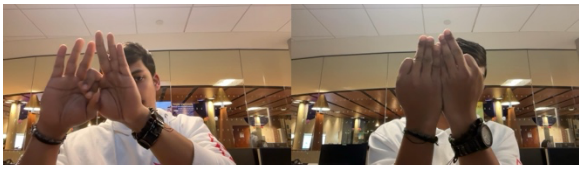
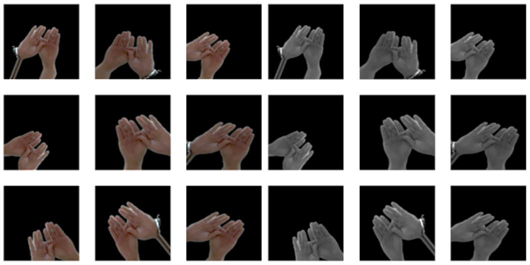


Fig 9. Raw Image with Background Noise

Step 2: Image Processing - Segmentation and Background Removal

To improve recognition accuracy, we introduced preprocessing techniques such as background segmentation as shown in Fig 10.a, thresholding, and color filtering as shown in Fig 10.b. This step removed much of the irrelevant background information, allowing the model to focus on hand features. By using these methods, we observed an improvement in classification accuracy and a reduction in false detections. However, challenges remained, as background segmentation algorithms were prone to errors under dynamic lighting conditions, and hand edges sometimes merged with shadows or objects in the environment. These erroneous segmentations become more significant with different angles and orientations paired with dynamic lighting condition. While this approach was better than raw image training, inconsistencies in segmentation prevented it from being the optimal solution.



a. Segmented Image b. Gray scaled Image

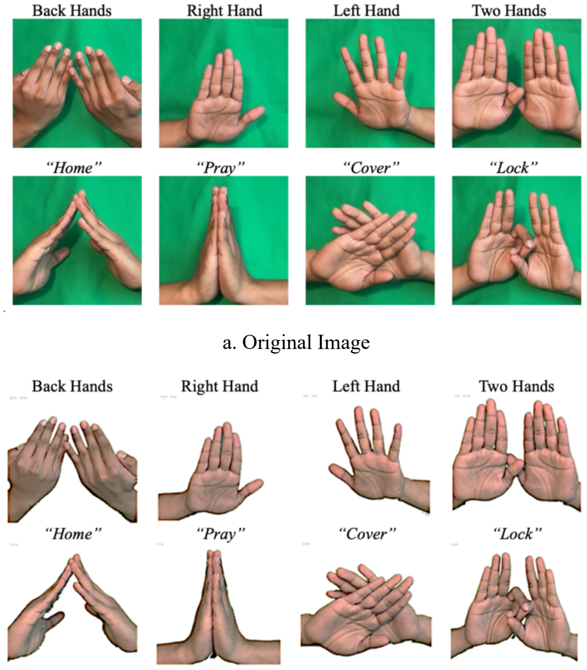
Fig 10. Background Removed Image with Augmentation [“*Lock”*]

Step 3: Implementing Green Screen for Enhanced Recognition

To further enhance accuracy, we implemented a green screen-based approach, providing a uniform background for training and recognition. By ensuring that the background remains consistent, the model could focus entirely on the hand features, eliminating distractions caused by environmental factors. This step resulted in a significant improvement in classification performance, reducing misclassification rates and ensuring reliable recognition even in challenging conditions. The green screen approach also optimized training efficiency by standardizing dataset conditions, making it the best approach among the three.

While the green screen method provides a clean and controlled background for gesture recognition, its impact on accuracy is minimal compared to standard background removal techniques. Both green screen and background-removed images (as shown in fig 11) yield similar recognition accuracy, meaning that the additional preprocessing step does not significantly enhance the model’s performance. However, the use of background removal introduces excessive computation delays, making the system slower and less efficient for real-time applications. The additional processing time required for background segmentation results in increased latency, ultimately affecting the overall responsiveness of the ASL recognition system. Given that accuracy remains nearly the same, the trade-off between computation time and performance makes background removal a less practical choice, especially when optimizing for real-time efficiency.

For training the CNN model in Layer 2, We created a dataset of two-hand ASL gestures using own hands under varying conditions to simulate diversity. Although the dataset was based on a single user, We introduced variations in hand orientation, lighting conditions, and hand positioning to help the model generalize better. To further enhance performance, data augmentation techniques were applied, including mirroring, rotation, and scaling. Rotation adjustments accounted for natural variations in hand positioning, making the model less sensitive to minor deviations in angle.



a. Original Image

b. Background Removed Image

Fig11.Layer 2 Dataset

Fig 11 describes images used for Layer 2 Training, where fig 11.a gestures captured with using of green screen and later used for training the model.

Additionally, to ensure the model could differentiate between intended gestures and unintentional hand movements, we used custom dataset containing various hand positions such as right hand, left hand, no hands, two hands, and back hands were introduced as shown in (fig11 first row). By first training the model to recognize the presence of hands and their patterns, it was better equipped to classify two hand gestures accurately. This preprocessing step enabled the model to first identify the number of hands present in the frame before proceeding to gesture classification. By incorporating a “no hands” category, the system reduced false positives caused by noise or unintentional movements. The hand detection model learned patterns in positioning and orientation, ensuring that the classification step only occurred when valid hands were detected.

During classification, the system follows a two-step verification approach: First, it verifies if two hands are present in the frame using the trained hand-detection model. Second, if both hands are detected, the system classifies the gesture by matching it with the most probable ASL sign in the trained dataset. This hierarchical classification ensures that only valid two-hand gestures are processed, minimizing misclassification due to partial hand visibility or incorrect posture.

For two hand static gestures, the system was tested using a pre-recorded video to simulate real-world ASL interactions. During testing, the system analyzed video input with frame extraction at 10 FPS (frames per second), simulating real-world ASL interactions. Our trained neural network model first verified hand presence and configuration, ensuring that at least two hands were present before attempting gesture classification. Once detected then model processed the frame to classify the specific ASL sign being performed. This method significantly reduced false detections by eliminating gestures detected from a single hand or irrelevant movements. Additionally, the model’s accuracy improved through pre-training with different hand positions, ensuring robustness in varying lighting conditions and backgrounds. The video was processed by extracting frames at a rate of 10 FPS, ensuring efficient detection while reducing redundancy. Each extracted frame was passed through the trained CNN model, which classified the gestures. To prevent repetitive gesture recognition, the system stores the last detected gesture and compares it with the newly recognized gesture. If the new detection is different from the previous one, it is printed and read aloud. This mechanism ensures fluid ASL-to-text conversion without unnecessary repetition, improving the system’s usability for real-time ASL communication.

To ensure robustness and reliability in real-time ASL recognition, our system incorporates fallback mechanisms at various stages of detection and classification. These fallback strategies are designed to handle scenarios where primary recognition methods encounter errors due to occlusions, misclassification, lighting variations, or background noise.

For single-hand gesture recognition (Layer 1), if Media Pipe fails to detect a hand in a frame, the system retries detection for a few consecutive frames before classifying it as a failure. This approach prevents temporary occlusions or poor lighting conditions from affecting recognition. Additionally, if a hand is detected but does not correspond to a predefined single-hand gesture, the system defaults to an "uncertain" state rather than forcing a misclassification.

In two hand gesture recognition (Layer 2), if the CNN model misclassifies a gesture, the system validates predictions over multiple consecutive frames before confirming the final classification. This prevents misclassifications caused by momentary hand overlaps or ambiguous hand positioning.

Finally, during gesture-to-text and gesture-to-speech conversion, the system ensures that redundant gestures are not repeatedly recognized and spoken out. If consecutive frames detect the same gesture, the system only registers a new output when a distinct gesture appears, reducing repetitive outputs and improving readability.

By integrating these fallback mechanisms, the ASL recognition system enhances accuracy, minimizes false detections, and ensures seamless real-time interaction, making it more reliable for real-world applications.

IV. RESULTS

The CNN model was able to recognize complex interactions between hands, overcoming the limitations of landmark-based methods. The use of green screen preprocessing significantly improved gesture segmentation, ensuring that the CNN model focused solely on hand structures rather than background distractions.

The foot pedal confirmation mechanism was tested for gesture validation accuracy and user experience. It effectively reduced false positives, ensuring that only intended gestures were stored or translated. The error rate for gesture confirmation was less than 1%, significantly improving system reliability compared to traditional gesture-based confirmation methods, which often resulted in accidental activations.

The real-time evaluation of the system demonstrated efficient and accurate gesture recognition for both single-hand and two hand ASL gestures. For Layer 1, word formation was optimized by implementing a confirmation-based mechanism using a foot pedal, ensuring that unintended gestures were not stored. The system successfully converted individual letter gestures into complete words, maintaining a high level of accuracy.

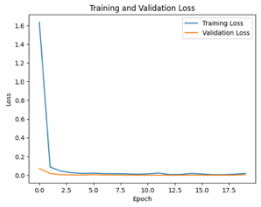
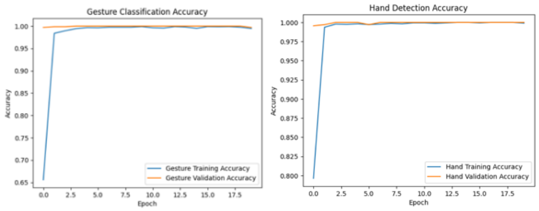


Fig 12. Training and Validation Loss



a. Gesture Accuracy b. Hand Detection Accuracy

Fig 13. Trained Model Accuracy metrics

For Layer 2, Figures 12 and 13 shows the training and validation loss results and accuracy metrics of two hand gesture recognition tested using video input with a frame rate of 10 FPS and test dataset. The model effectively detected gestures while minimizing redundant outputs by ignoring consecutive repetitions of the same gesture. This significantly improved usability, as only new gestures were printed and read aloud, preventing output spam and enhancing real-time communication accuracy.

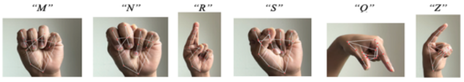
For single-hand static gestures (Layer 1 – Mediapipe-based detection), the system reliably recognized a total of 20 alphabetic signs, 9 digits, and 16 additional static ASL gestures made by right hand. Rather than relying on deep learning, this layer used distance and angle-based rule logic with real-time landmark tracking via Computer Vision Zone and MediaPipe, enabling quick and responsive detection. The real-time feedback on the hand landmarks allowed users to see instant detection, making the system efficient for practical use.

For two-hand static gestures (Layer 2 – CNN-based classification), the system was trained and tested on four ASL gestures. Although the training dataset used was limited, the model showed successful recognition of these signs in live demonstrations. To strengthen detection reliability, the model was first trained to differentiate between hand configurations such as "right hand," "left hand," "two hands," "back hands," and "no hands," before classifying the gesture itself. While full-scale accuracy testing wasn't performed, the results indicate that the model performs well for the trained classes and serves as a scalable foundation for future expansion.

V. Challenges and Limitations

Despite significant advancements in ASL gesture recognition, several challenges remain in two hand detection, real-time processing, dataset variability, and confirmation mechanisms. These limitations must be addressed to ensure high accuracy, efficiency, and reliability in practical applications.

1. Single hand Gesture Recognition Challenges



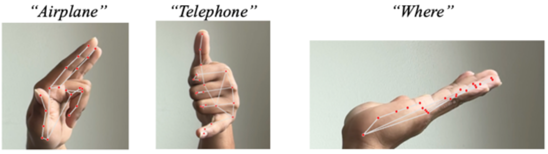


Fig 14. Unrecognizable Gestures

One of the major challenges faced in our ASL recognition system is the inconsistent detection of landmarks, particularly for certain hand gestures such as *"M", "N", "R", "S", "Q"* and *"Z"* as shown in Fig. 14. These gestures often experience landmark fluctuation, which leads to misclassification or complete recognition failure. The primary reason behind this issue is the close positioning of fingers in these signs, making it difficult for MediaPipe to accurately differentiate individual landmarks. Additionally, gestures like *"M", "N", "R"* and *"S"* involve multiple fingers being placed close together, which often results in occlusion, causing the system to misinterpret their exact placements. Similarly, for gestures such as "*Q"* and *"Z"*, the orientation of the hand leads to inconsistent edge detection, where the system struggles to correctly identify the finger outlines, leading to fluctuating landmark positions. These challenges highlight the limitations of our current landmark detection approach and emphasize the need for improvements in handling overlapping and occluded finger positions for more reliable gesture classification.

Like the challenges observed in ASL alphabet gestures, our system faced landmark detection inconsistencies with certain static signs, particularly *"Airplane", "Telephone"* and *"Where"* as shown in Fig 14. These gestures exhibited landmark fluctuations, leading to classification errors and recognition failures. One of the key reasons for these inconsistencies is the complex positioning of fingers in gestures like *"Airplane"* and *"Telephone"* where multiple fingers extend in different directions, making it challenging for MediaPipe to accurately detect individual landmarks. Additionally, the *"Where"* gesture, which involves a side-facing hand position, often results in landmark misalignment, leading to incorrect interpretations by the system. Another issue arises due to perspective and angle distortions, where these gestures are performed at an orientation that causes certain fingers to appear shorter or partially occluded. This affects the reliability of landmark tracking, making recognition unstable in real-time applications.

2. Two hand Gesture Recognition Challenges

One of the most complex challenges in ASL recognition is accurate two hand gesture classification. Unlike single-hand gestures, two hand gestures involve overlapping hands, occlusions, and dynamic interactions, making it difficult to extract clear landmark positions. Traditional landmark-based recognition methods, such as distance and angle measurements, often fail when hands intersect or partially cover each other, leading to incorrect classifications. Although CNN-based models improve two hand recognition, they require large datasets for proper training, increasing computational complexity and inference time.

3. Real-Time Processing Limitations

For ASL recognition to be practical in real-world applications, it must operate in real time with minimal delays. Many deep learning models require high computational power, making real-time execution on low power devices challenging. While Media Pipe based approaches efficiently handle single-hand static gestures, more complex gestures demand CNN processing, which is computationally expensive. This creates a trade-off between accuracy and speed, as higher accuracy often comes at the cost of increased processing time. Additionally, background processing and lighting conditions introduce further delays, making real-time implementation difficult without optimized preprocessing techniques such as green screen segmentation.

4. Dataset Variability and Model Generalization

ASL recognition models require diverse datasets to ensure generalization across different hand shapes, sizes, and skin tones. However, publicly available datasets often lack sufficient two hand examples, leading to biases in training models. Additionally, dataset inconsistencies such as variations in hand positioning, signer styles, and environmental lighting make it harder for models to generalize across real-world users. To mitigate this, data augmentation techniques such as rotation, flipping, and scaling are applied, but these do not fully replicate natural hand movement variations.

5. Background Noise and Lighting Conditions

Many vision-based ASL recognition systems are highly sensitive to lighting variations and background noise, affecting gesture detection accuracy. Traditional background removal techniques, such as Gaussian filtering and thresholding, often fail under dynamic lighting conditions, introducing segmentation errors. The green screen method used in this research significantly reduces background-related misclassification, ensuring consistent hand segmentation, but its effectiveness depends on controlled environments. In uncontrolled environments, external lighting changes can still impact detection performance, requiring further adaptive preprocessing techniques.

6. Gesture Confirmation and Human Bias

Another challenge in ASL recognition is the lack of reliable confirmation mechanisms. Many systems rely on gesture-based confirmation, such as double gestures or head tilts, but these methods introduce human bias and misinterpretation errors. For instance, accidental hand movements may trigger incorrect detections, causing frustration for users. To address this, our system incorporates a foot pedal switch, which ensures clear and deliberate confirmation, eliminating ambiguity and bias. While this method significantly improves gesture validation, it may require additional user training for effective utilization in real-world scenarios.

7. Hardware and Software Constraints

Although the ASL recognition system is designed to function on standard computing hardware, deep learning models require GPU acceleration for optimal performance. While CPU based execution is feasible, it results in longer inference times, making real-time implementation difficult on low end devices. Additionally, software dependencies such as OpenCV, MediaPipe, and TensorFlow require periodic updates, which may introduce compatibility issues across different systems.

VI. Conclusion

This research introduces a multi-layered ASL recognition system that integrates MediaPipe for single-hand static gestures and a CNN-based deep learning model for two-hand static gestures, addressing key challenges related to gesture recognition, background noise, and real-time processing. The implementation of green screen preprocessing and a foot pedal confirmation mechanism has significantly improved classification accuracy, reduced misclassification errors, and enhanced real-time usability.

Moving forward, future work will focus on addressing the limitations in the 6 letter gestures and 3 static signs by new feature extraction. Multi hand gestures by refining their recognition through dataset expansion, improved preprocessing techniques, and adaptive neural network training. Additionally, the system will be expanded to include all remaining two-hand gestures, ensuring a more comprehensive ASL recognition framework. Another crucial enhancement will be the integration of AI-driven contextual analysis, which can further improve sentence formation and natural ASL translation, making the system more practical for real-world communication.

Overall, this research contributes to the development of a highly efficient, accessible, and scalable ASL recognition system, enabling better human-computer interaction and assistive communication for the DHH community. By combining machine learning, computer vision, and innovative preprocessing techniques, this system provides a reliable and effective solution for ASL interpretation, paving the way for more inclusive and intelligent digital communication tools.

VII. REFERENCES

Bibliography

[1 G. A. Edge, "MediaPipe, “Hand Landmarker"," Google, [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker.

[2 S. S. Shivashankara S, "American Sign Language Recognition System: An Optimal Approach," International Journal of Image, Graphics and Signal Processing(IJIGSP), vol. 10, 2018.

[3 S. G. K. K. a. J. K. P. Molchanov, "Hand Gesture Recognition with 3D Convolutional Neural Networks," in IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

[4 A. Aggarwal, "American Sign Language Recognition using Computer Vision and Deep Learning," 2023.

[5 M. A. A. M. M. Mrs.M.Stella Inba Mary, "HAND GESTURE RECOGNITION USING MEDIAPIPE AND OPENCV," International Journal of creative research Thoughts, no. ISSN: 2320-2882, 2024.

[6 J. A. S. Lakuntara Pallahidu, "A Real-Time Hand Gesture Recognition System for Converting Sign Language to Alphabetic Character Using Deep Learning Approach," in Brawijaya International Student Conference (BISC), Malang City, Indonesia, 2022.

[7 V. Mudgal, "Real-Time Gesture Recognition Using MediaPipe Hands," [Online]. Available: https://mudgalvaibhav.medium.com/real-time-gesture-recognition-using-googles-mediapipe-hands-add-your-own-gestures-tutorial-1-dd7f14169c19.

[8 T.-W. Chong and B.-G. Lee, "American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach," Sensors, 2018.

[9 R. O. S. J. A. C. T.-a. M. A. B. P. J. R. R. F. a. X. J. O. G. Lean Karlo S. Tolentino, "Static Sign Language Recognition Using Deep Learning," International Journal of Machine Learning and Computing, 2019.