American Sign Language Detection

Mahdi Jafarkhani [1] **Harish Renganathan** [1]

Abstract

Individuals who are deaf and mute often face significant communication barriers, relying on sign language, which involves specific hand shapes and movements. However, this mode of communication is not widely understood by those who primarily use spoken and written languages, highlighting the need for technological solutions. Although extensive research has addressed sign language recognition in many global languages, there remains an opportunity for the development of tools tailored to local dialects. This paper introduces a real-time system for recognizing American Sign Language (ASL) gestures, leveraging advanced computer vision and machine learning techniques, with feature extraction facilitated by the Mediapipe library.

1. Introduction

Sign language is a complex form of communication that relies heavily on visual cues rather than auditory signals. With advancements in technology, there has been a significant focus on developing systems that can bridge the communication gap between those who use sign language and those who do not understand it. While traditional sign language recognition systems have relied on hardware such as gloves and sensors, recent developments in computer vision and machine learning offer more flexible and accessible solutions.

Real-time processing of sign language is particularly critical in ensuring smooth and effective communication. The ability to recognize gestures as they happen can lead to significant improvements in tools for education, translation, and everyday communication for the deaf community. However, real-time processing presents its own set of challenges, particularly in terms of computational efficiency and accuracy. This paper explores the implementation of a real-time ASL recognition system that leverages modern computer vision tools and machine learning models, with an emphasis on the balance between accuracy and processing speed.

American Sign Language (ASL) is not only a tool for communication but also a representation of the cultural identity of the deaf community in North America. The development of tools that can accurately and efficiently recognize ASL gestures is a step towards greater inclusion and accessibility in society. This paper contributes to this goal by presenting a novel approach to ASL recognition that combines the robustness of MediaPipe with the power of machine learning models such as Random Forest classifiers.

2. Related Work

The field of sign language recognition has seen a wide range of approaches, from hardware-based systems to advanced machine learning models. Rupesh Kumar et al (2). focused on using the Mediapipe library combined with a Convolutional Neural Network (CNN) for real-time ASL gesture recognition. Their system achieved high accuracy, emphasizing its utility for communication devices for individuals with hearing impairments. The approach demonstrated effectiveness in detecting all ASL alphabets, with the potential for application in other sign languages with similar hand movements.

Similarly, Praiselin et al (1). utilized Mediapipe for feature extraction, specifically targeting ASL alphabets and numeric digits. Their system was designed for real-time operation, employing modern computer vision and machine learning techniques. This work aimed to enhance accessibility for the deaf and hard-of-hearing community through advanced gesture recognition technologies.

Another work that contributes to this problem is "Gesture-Recognition" repository (3), which is focused on recognizing ASL gestures using pre-trained MediaPipe models. It identifies hand landmarks via the BlazePalm Detector and computes Euclidean distances between key points on the hand to classify ASL letters using a Bayesian classifier. The repository includes scripts for exploring data, training the classifier, and on video feeds.

[*]Equal contribution [1]Department of Computer Science, University of Stuttgart, Stuttgart, Germany. Correspondence to: Mahdi Jafarkhani <mahdi.jafarkhani@ians.uni-stuttgart.de>, Harish Renganathan <rharish4444@gmail.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Research in sign language recognition has evolved significantly over the years. Early systems often relied on cumbersome hardware, such as data gloves or motion capture systems, to detect and interpret hand gestures. While these systems were effective, they were not practical for everyday use due to their complexity and cost. More recent approaches have shifted towards vision-based systems, which use cameras to capture hand movements and machine learning algorithms to interpret them.

Studies have demonstrated the utility of the Mediapipe library for real-time hand tracking and feature extraction in ASL recognition systems, highlighting its effectiveness in enhancing gesture recognition accuracy. Mediapipe's ability to perform real-time, high-fidelity tracking of hand landmarks makes it an ideal tool for this application. Research has shown that CNNs are highly effective in classifying ASL alphabets and digits, providing high accuracy in gesture recognition by capturing intricate hand movements and shapes.

The evolution of machine learning techniques has also played a crucial role in advancing sign language recognition. Early systems often used simpler models, such as Support Vector Machines (SVMs) or Hidden Markov Models (HMMs), to classify gestures. While these models were effective to some extent, they struggled with the complexity and variability of hand gestures in real-world scenarios. The advent of deep learning, particularly CNNs, has significantly improved the accuracy and robustness of sign language recognition systems. CNNs are capable of learning complex, hierarchical representations of hand gestures, making them well-suited for this task.

3. Background

Definition (American Sign Language (ASL)): *American Sign Language (ASL) is a natural language primarily used by deaf communities in the United States and parts of Canada. It employs a visual-gestural mode of communication, utilizing hand shapes, facial expressions, and body postures to convey meaning. (Figure 1)*

Definition (MediaPipe): *MediaPipe is an open-source framework developed by Google Research for building cross-platform machine learning pipelines. It provides tools and pre-trained models for tasks such as object detection, pose estimation, and gesture recognition, making it suitable for real-time applications on mobile devices and other platforms.*

3.1. Dataset

The ASL dataset on Kaggle for alphabets (4) and numbers (5) is a comprehensive collection of images representing various signs in American Sign Language. It covers over 34,000 images corresponding to different ASL signs, providing a robust foundation for training machine learning models for gesture recognition and translation tasks. Annotating this dataset involved significant challenges, such as dealing with variations in hand orientation, lighting conditions, and background noise. These challenges were mitigated through careful pre-processing and augmentation techniques, such as normalizing the images, applying random rotations, and adjusting brightness and contrast. This ensures that the models can learn to recognize gestures in a variety of conditions. (Figure 2)

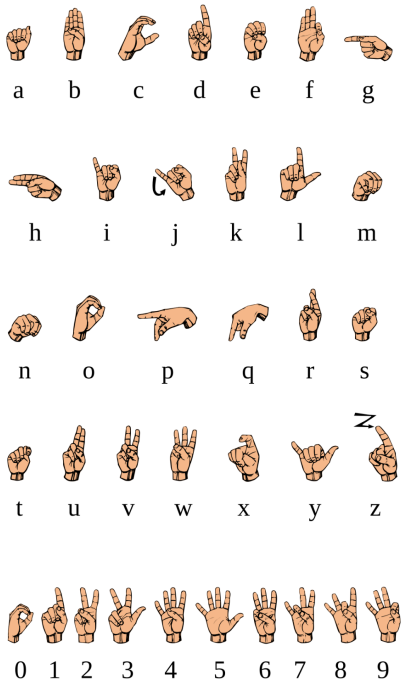

Figure 1. **ASL Alphabet and digits, from Wikipedia**

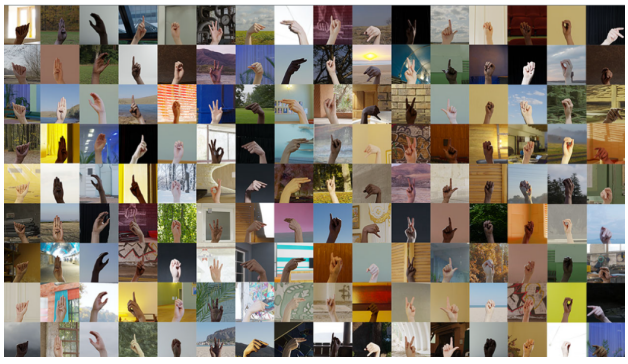

Figure 2. **Sample images from Kaggle (4)**

3.2. MediaPipe

MediaPipe originated in the early 2010s as part of Google's efforts to advance machine learning and computer vision technologies. Its initial application dates back to 2012, when it was employed to perform real-time analysis of video and audio content on YouTube.

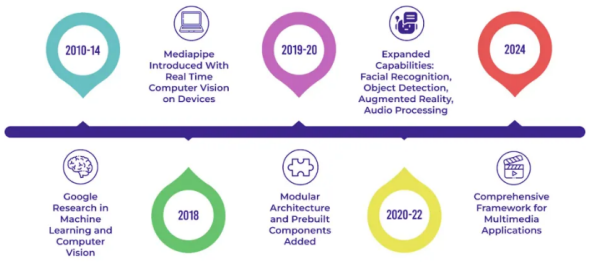

Figure 3. **A Timeline of MediaPipe's Progress**

This framework is particularly suited for our project on American Sign Language (ASL) recognition, especially its robust capabilities in computer vision and machine learning.

3.3. MediaPipe Framework

MediaPipe, an open-source framework developed by Google, has revolutionized the way real-time machine learning pipelines are built. In the context of ASL recognition, MediaPipe provides the crucial ability to detect and track hand landmarks with high accuracy and low latency. The framework's modular design allows developers to easily integrate different components, such as hand detection and gesture recognition, into a cohesive pipeline.

Each hand is represented by 21 landmarks as shown in Figure 4, which are the key points used to understand the structure and position of the hand in 3D space. These landmarks include crucial points like the tips of the fingers, the knuckles, and the base of the palm. By analyzing the relative positions of these landmarks, it is possible to derive features that can be used to distinguish between different ASL gestures.

The feature extraction process involves calculating Euclidean distances between selected pairs of landmarks and angles formed by triplets of points. These features capture the spatial configuration of the hand, which is essential for distinguishing between gestures. The choice of these specific features was guided by the need to balance accuracy with computational efficiency, ensuring that the system could operate in real-time.

There are 3 main features which are extracted in our pipeline:

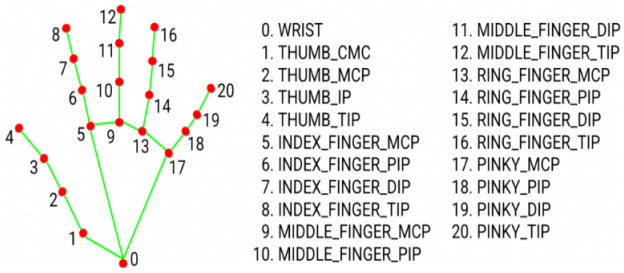

Figure 4. **MediaPipe's Landmark**

- **Calculating distances of two landmarks:** We scale the coordinates and calculate the Euclidean distance of 9 pairs of landmarks as our features, and classify the detected hand to one of the labels.

  From Figure 4, we can calculate the following distances (from 21 landmarks): 4-0, 8-0, 12-0, 16-0, 20-0, 4-8, 8-12, 12-16 and 16-20.

- **Calculating angles between landmarks:** We also added the angles between 4 triplets of landmarks, to increase accuracy and reduce confusion. Once again from Figure 4, we can extract the following angle: 4-0-8, 8-0-12, 12-0-16 and 16-0-20

- **Detecting palm state:** Since ASL alphabets and digits are palm-aware, which means that whether the palm of your hand is facing the camera or not is important, we also include that as a binary feature.

The framework's modular design allows us to construct pipelines using pre-built components known as calculators. These calculators can be customized and connected to handle tasks such as hand gesture detection, finger tracking, and pose estimation, which are essential for interpreting ASL gestures accurately.

MediaPipe optimizes performance through hardware acceleration (e.g., GPU) and efficient algorithms, ensuring low-latency processing. This capability is vital for real-time ASL interpretation, enabling immediate feedback for users.

We can integrate TensorFlow Lite models into MediaPipe, enabling us to deploy trained neural networks for recognizing ASL signs directly on mobile and embedded devices. This is more towards the future work.

4. Workflow

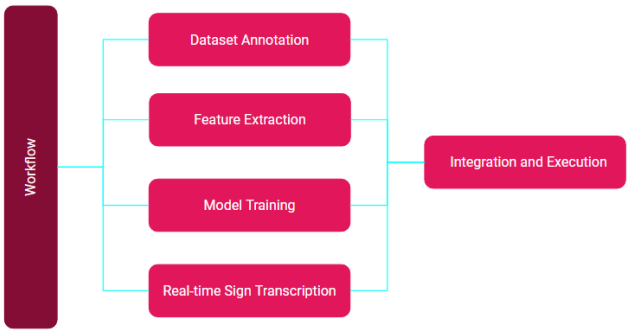

Figure 5. **Workflow**

The implementation involves several steps, including dataset annotation, feature extraction, model training, and real-time sign transcription. Below, we detail each component and its role in the overall system.

4.1. Dataset Annotation

The first step in our project was to annotate a dataset of hand images with key points representing the positions of various hand landmarks. We developed a class, `DatasetAnnotator`, to automate this process. This class leverages the HandLandmarker from Mediapipe to detect hand landmarks from images stored in the dataset folder.

The `DatasetAnnotator` class iterates through each class of images, processes each image to extract hand landmarks, and stores these landmarks in CSV files. This ensures that our dataset is well-annotated and ready for the subsequent feature extraction phase. The progress of this annotation process is tracked and displayed using a custom progress bar function, enhancing the usability and monitoring of the dataset annotation.

4.2. Feature Extraction

Once the dataset is annotated, the next step is to extract meaningful features from the annotated key points. We created a `FeaturesExtractor` class for this purpose. This class reads the key point data from CSV files and computes various geometric features, such as distances and angles between key points.

The `getdistance` and `getangles` functions in the `util.py` module calculate these features. Distances between significant landmarks provide spatial information, while angles between connected landmarks offer insights into the hand's posture.

4.3. Model Training

With the extracted features, we proceeded to train machine learning models to classify ASL alphabet signs. The `LandmarkDetector` class handles this aspect of the implementation. We employed two types of models: a Random Forest classifier and a simple Deep Neural Network (DNN).

The Random Forest classifier is trained using scikit-learn, leveraging the extracted features to learn patterns associated with each ASL sign. The training process involves splitting the dataset into training and testing sets, fitting the classifier, and evaluating its performance. The trained model is then saved for later use.

In parallel, we also prepared a DNN model using TensorFlow Keras, depicted in Figure 6. The DNN processes the feature vectors to learn more complex patterns that might be missed by traditional classifiers. This model is saved and can be loaded for real-time predictions.

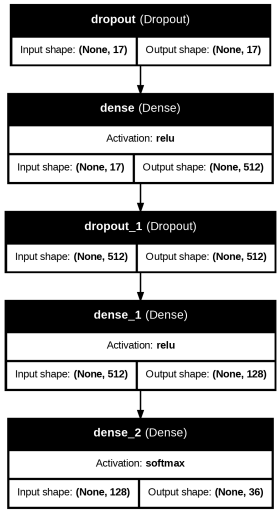

Figure 6. **Structure of the DNN**

4.4. Real-time Sign Transcription

The final step in our implementation is the real-time transcription of ASL signs from video input. The `SignTranscriber` class is designed for this purpose. It captures video from a webcam, processes each frame to detect hand landmarks using Mediapipe, and then uses the trained models to predict the corresponding ASL sign.

The transcription process involves capturing each video frame, detecting hand landmarks, and extracting features. These features are then fed into the trained classifier or DNN to predict the ASL sign. The predicted sign is displayed on the video feed, providing immediate feedback to the user.

4.5. Integration and Execution

The entire system is integrated and executed through the `main.py` script. This script orchestrates the different components, starting from dataset annotation (if needed), feature extraction, model training, and finally, launching the real-time transcription interface.

By structuring the project in this modular fashion, we ensured that each component could be developed, tested, and improved independently. This modularity also facilitates future enhancements, such as incorporating more sophisticated models or expanding the dataset to include additional signs or dynamic gestures.

This ASL detector project demonstrates the effective use of computer vision and machine learning techniques to address a practical problem. By leveraging Mediapipe for landmark detection and combining it with robust feature extraction and classification methods, we achieved a system capable of recognizing ASL alphabet signs in real-time. The implementation details provided here serve as a comprehensive guide to understanding the inner workings of the project and form a solid foundation for further development and refinement.

5. Random Forest classifier and DNN: A Comparative Study

In this project, we compared the performance of two machine learning models: a Deep Neural Network (DNN) and a Random Forest classifier. Both models were trained on the same dataset of ASL hand gestures, to evaluate their effectiveness in real-time gesture recognition.

5.1. Model Performance

The DNN, implemented using TensorFlow Keras, was designed to capture complex patterns in the hand gesture data by processing the feature vectors extracted from the hand landmarks. The model consisted of several dense layers followed by fully connected layers, enabling it to learn hierarchical representations of the input data.

On the other hand, the Random Forest classifier, implemented using scikit-learn, utilized the extracted features to build an ensemble of decision trees. This model is known for its robustness and ability to handle noisy data, making it a good candidate for initial testing.

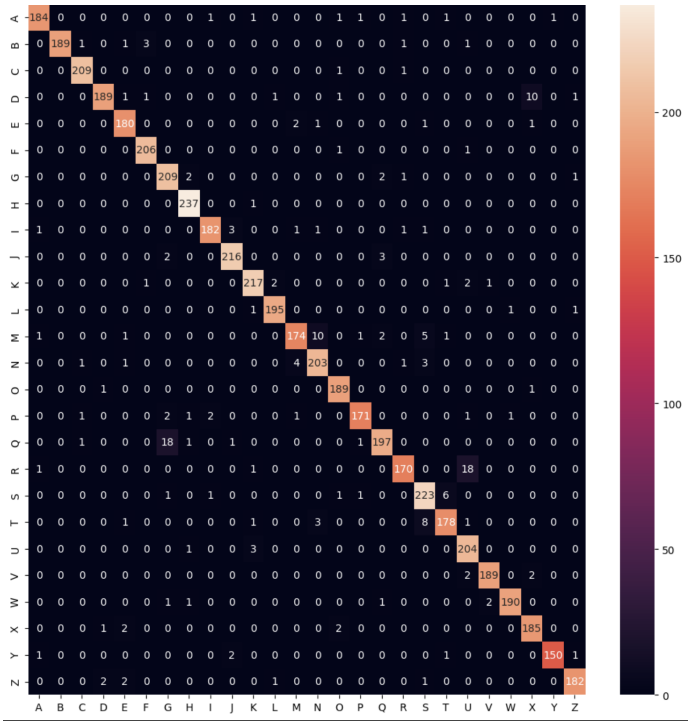

Figure 7. **Confusion Matrices for DNN model**

As shown in the confusion matrices in Figure 7, clearly shows considerable confusion in similar ASL digits or numbers. For example, between letter **U** and letter **R**, or between letter **Q** and letter **G** we notice some miss-classifications. This matrix helped improve the feature engineering part of the project and enhanced our approach. In general, the Random Forest model achieved higher overall accuracy, particularly in distinguishing between similar gestures, due to its ability to capture subtle variations in the data. Also, the Random Forest model exhibited faster training times and was less computationally intensive during inference, making it more suitable for deployment on resource-constrained devices.

6. Challenges and Solutions

Developing a real-time ASL recognition system comes with numerous challenges, each requiring careful consideration and innovative solutions. One of the primary challenges encountered was dealing with noisy data, particularly images where the hand was partially obscured or incorrectly detected by the MediaPipe framework. To address this, we implemented a data cleaning step that filtered out low-confidence detections, ensuring that only high-quality data was used for training.

Another significant challenge was maintaining real-time performance. The initial implementation of the DNN model,

while accurate, was too slow for real-time applications. We optimized the model by reducing the depth of the network and implementing techniques such as model pruning and quantization. These optimizations reduced the inference time significantly, making the system viable for real-time use.

7. Real-World Applications and Future Work

7.1. Practical Applications

The ASL recognition system developed in this project has numerous potential applications. One of the most immediate is in assistive devices, where the system could be integrated into wearable technology, such as smart glasses, to provide real-time translation of sign language into spoken or written text. This would be particularly beneficial in educational settings, enabling smoother communication between deaf students and their hearing peers or teachers.

Another promising application is in the realm of augmented reality (AR) and virtual reality (VR). The system could be incorporated into AR glasses to provide real-time overlays of translated text when someone is signing, enhancing accessibility in various social and professional settings.

7.2. Future Enhancements

Future work on this project could focus on several areas for improvement. One major enhancement would be to expand the dataset to include dynamic gestures, which represent words or phrases rather than just individual letters. This would make the system more versatile and capable of handling more complex communication tasks.

Additionally, incorporating advanced deep learning models, such as transformer networks, could improve the system's accuracy, particularly for subtle or ambiguous gestures. Exploring the integration of this system with speech recognition technologies could also open up new possibilities for creating more comprehensive communication tools.

Also, since this project was focused only on images, it could be also a potential improvement to incorporate motions, since ASL is not only about static hand gestures, rather it covers hand motions. For example, to indicate letters **J** or **Z** the hand should be moved in a particular way, which is not supported in our work. It is also possible to incorporate both hands in our model, since some ASL pre-defined sentences like **Hello**, or **I love you** are transcribed with both hands.

The project's repository is publicly available on GitHub (6), allowing anyone to contribute further.

References

[1] Honesty Praiselin, E. J., Manikandan, G., Veronica, V., & Hemalatha, S. (2024). Sign language detection and recognition using MediaPipe and deep learning algorithm. *International Journal of Scientific Research in Science and Technology, 11*(2), 123-130. https://doi.org/10.32628/IJSRST52411223

[2] Kumar, R., Bajpai, A., Sinha, A., and Singh, S. K. Mediapipe and CNNs for real-time ASL gesture recognition. Department of CSE, Galgotias College of Engineering and Technology, AKTU, Greater Noida, India.

[3] https://github.com/aqua1907/Gesture-Recognition

[4] https://www.kaggle.com/datasets/lexset/synthetic-asl-alphabet

[5] https://www.kaggle.com/datasets/lexset/synthetic-asl-numbers

[6] https://github.com/M-Jafarkhani/ASLTranscriber