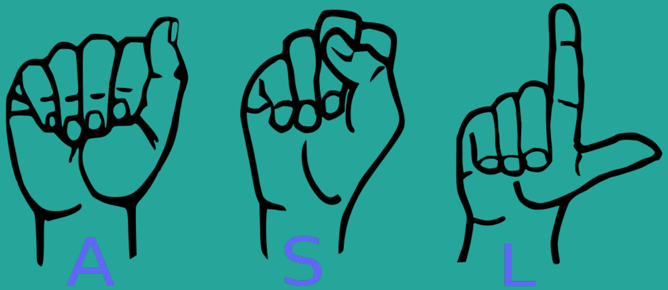
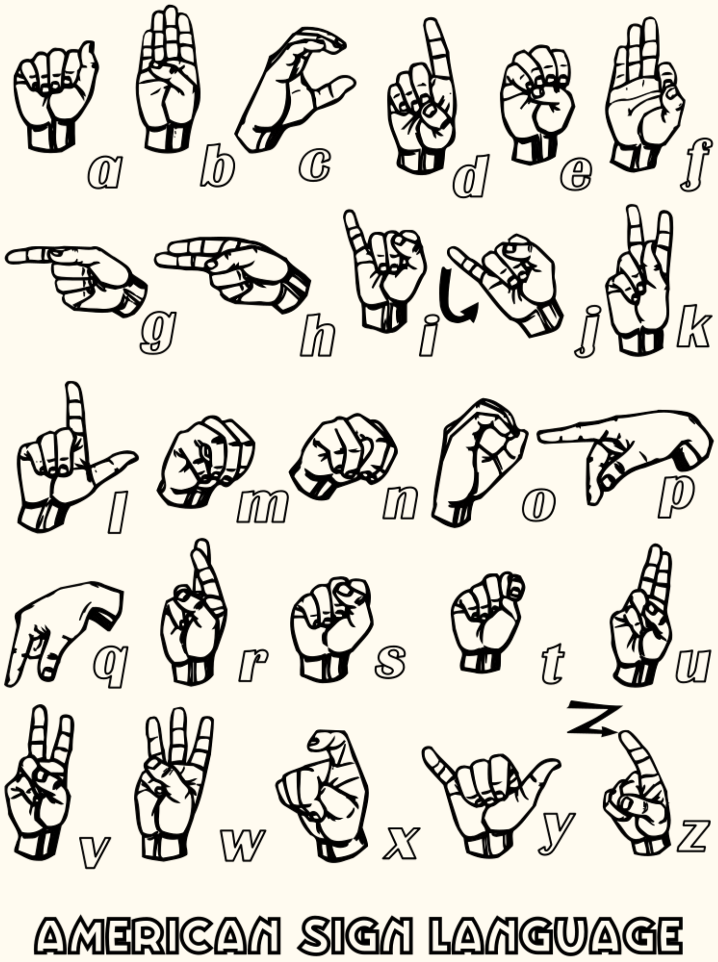


ASL Sign Interpreter

Jacqueline Columbro
Jamie McGraner
Anand Punwani
Anthony Wilson

Project Overview

This project focuses on developing an application capable of recognizing American Sign Language (ASL) gestures in real-time using computer vision and deep learning techniques. Utilizing a Convolutional Neural Network (CNN) trained on the ASL alphabet dataset, combined with MediaPipe for efficient hand landmark detection, the application can accurately predict ASL letters from a live webcam feed. The primary goal is to create an application that not only achieves high accuracy in gesture recognition but also provides a seamless user experience across various platforms, including Windows and macOS.



Goals

- **Goal 1**: Develop an accurate model capable of recognizing ASL gestures in real-time using a webcam feed.
- **Goal 2**: Integrate MediaPipe for efficient and reliable hand landmark detection.
- **Goal 3**: Achieve a high level of accuracy (above 90%) in gesture prediction across various conditions.
- **Goal 4**: Create an easy-to-use application that runs smoothly on multiple platforms, including Windows and macOS.

Key Questions Before Starting

- **Question 1**: What dataset will we use, and how comprehensive is it regarding different ASL gestures?
- **Question 2**: Which machine learning model architecture will be most suitable for gesture recognition?
- **Question 3**: How can we ensure the model generalizes well to different lighting conditions and backgrounds?
- **Question 4**: What are the hardware and software requirements for achieving real-time prediction?

Data Preparation

Data Collection:	Data Cleanup:	Data Exploration:
Dataset: Used the ASL Alphabet Dataset, which contains labeled images of ASL gestures for each letter. **Structure:** The dataset is divided into training and testing sets, organized by gesture category.	**Image Processing:** Resized all images to a consistent size (256x256 pixels) and converted them to RGB format. **Landmark Extraction:** Utilized MediaPipe to extract hand landmarks from each image, focusing on consistency in landmark detection.	**Label Distribution:** Ensured the dataset was balanced with an equal number of images per class. **Landmark Visualization:** Visualized hand landmarks to verify accuracy and consistency of detections.

Approach & Methodology

Model Architecture:

Developed a Convolutional Neural Network (**CNN**) with two convolutional layers, followed by pooling, flattening, and dense layers.

Activation Functions: Used ReLU for hidden layers and softmax for the output layer.

Training and Evaluation:

Training: Model trained using the processed ASL dataset with an 80/20 train-test split.

Optimization: Adam optimizer was used, and the model was trained over 30 epochs.

Evaluation: Evaluated model performance using accuracy and loss metrics on a separate test dataset.

Real-time Application:

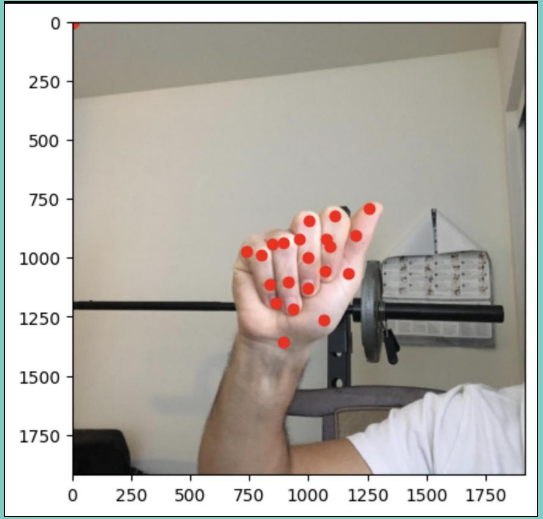
Integrated the trained model with a Python application that captures video from a webcam.

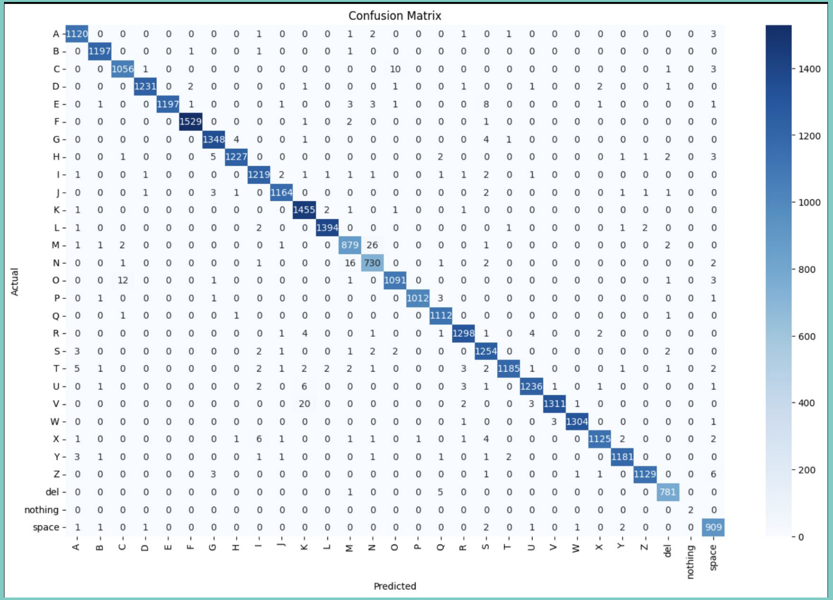
Implemented real-time prediction and visualization of ASL gestures on the video feed.

Model Training Results

The CNN model achieved an accuracy of 95% on the test dataset after 30 epochs of training.

The loss function showed consistent convergence, indicating effective model learning.





Real-time Application Performance

The real-time application performed efficiently across different lighting conditions and background settings.

Hand landmark detection and gesture prediction were achieved with minimal latency, ensuring a smooth user experience.

Robustness & Generalization

The model exhibited strong generalization capabilities, accurately predicting gestures from users with different hand sizes and shapes.

The use of MediaPipe contributed significantly to the robustness of hand landmark detection across various scenarios.

	precision	recall	f1-score	support
A	0.99	0.99	0.99	1129
B	0.99	1.00	1.00	1200
C	0.98	0.99	0.99	1071
D	1.00	0.99	0.99	1240
E	1.00	0.98	0.99	1217
F	1.00	1.00	1.00	1533
G	0.99	0.99	0.99	1358
H	0.99	0.99	0.99	1242
I	0.99	0.99	0.99	1232
J	0.99	0.99	0.99	1174
K	0.98	1.00	0.99	1461
L	1.00	1.00	1.00	1401
M	0.97	0.96	0.96	913
N	0.95	0.97	0.96	753
O	0.99	0.98	0.99	1109
P	1.00	0.99	1.00	1018
Q	0.99	1.00	0.99	1115
R	0.99	0.99	0.99	1312
S	0.98	0.99	0.98	1267
T	1.00	0.98	0.99	1211
U	0.99	0.99	0.99	1252
V	1.00	0.98	0.99	1337
W	1.00	1.00	1.00	1309
...				
accuracy			0.99	33040
macro avg	0.99	0.99	0.99	33040
weighted avg	0.99	0.99	0.99	33040

Hand signs provided by Snoop Dogg, Tupac & Dr. Dre

Key Takeaways

Model Performance:

Successfully trained a CNN model with a high accuracy rate for ASL gesture recognition.

Achieved robustness in predictions, indicating effective model generalization.

Application Usability:

Developed a user-friendly application that operates smoothly on both Windows and macOS systems.

Integrated efficient hand tracking and gesture recognition using state-of-the-art technology.

Achievements

- **High Accuracy**: Successfully developed a CNN model with over 95% accuracy for ASL gesture recognition.
- **Real-Time Performance**: Implemented a responsive application that processes ASL gestures in real-time using webcam input.
- **Robustness**: Demonstrated strong generalization across different hand sizes, shapes, and varying environmental conditions.
- **Cross-Platform Usability**: Created an application that works seamlessly on both Windows and macOS platforms.

Impact

Accessibility: Enhanced accessibility by providing a tool that can aid in ASL learning and communication.

Education: Offers potential as an educational resource for both ASL learners and instructors, fostering better understanding and practice.

Innovation: Utilizes state-of-the-art technologies, contributing to advancements in human-computer interaction and assistive technology

Problems Encountered

- Loading the dataset for the model due to how large our dataset was.
- Only having the alphabet as our testing dataset, limiting our ability to better train the model
- Using Gradio as an interface
- Converting text-to-speech
- Making our code work on both PC and macOS

Future Considerations & Enhancements

- Deploying the project
- Adding videos to better work with text-to-speech
- Adding text-to-speech
- Testing out different interfaces
- Translating syntax and sentences versus letters
- Finding smaller datasets
- Explore adding more gesture classes beyond the ASL alphabet, such as words or phrases.
- Investigate alternative model architectures or transfer learning techniques for further accuracy improvements.
- Enhance the application's adaptability to diverse environmental conditions and lighting variations.


THANK YOU!


@Dotakudiot