

### Project Summary

#### What is CAMASL?

**CAMASL** is a web application that will help users to learn the **ASL alphabet** using a machine learning model to predict what sign the user is performing and give a score based on **how accurately the sign was performed**.

#### How it Works:

- The user will **login or register** to my NextJS web application and will be redirected to **the main page**.
- The user can pick from a range of **random ASL signs to perform** and when they are ready to perform, they can hit the **start session button** which will start tracking the users' hands. This is done using the open-source OpenCV libraries that **track landmarks on the users' hands**.
- A function within the web app will **capture the frames of the users' hand** and pass it into the model. This will **generate and save scores** for the user scores page that is saved along with their login info **in a MongoDB database**

### Model Training

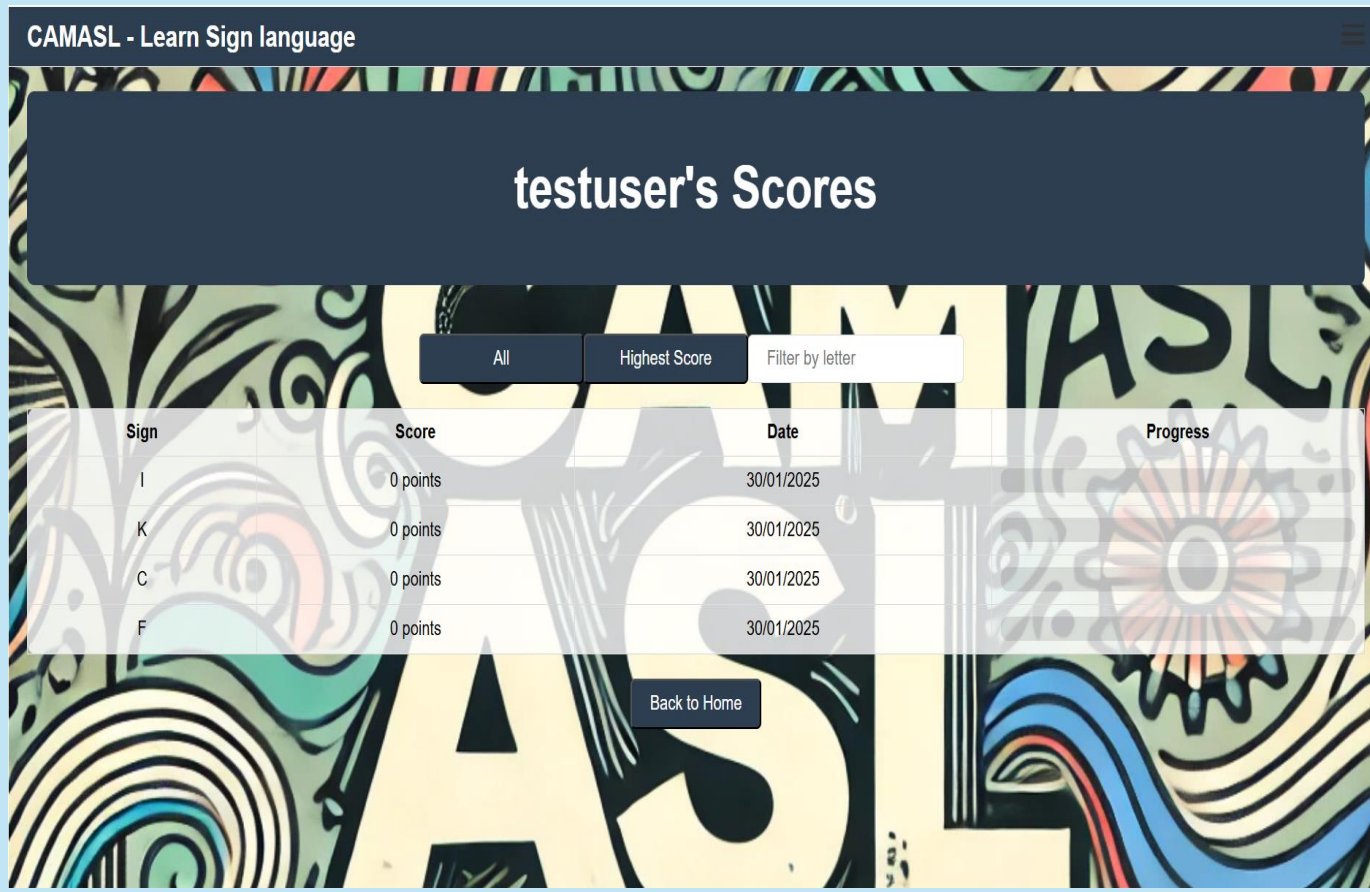

The model was trained using a process called **transfer learning** that is used to leverage existing knowledge from another pre-trained model for new tasks.


Approximately **11,000 images** are used to train the model. Each image contains hand landmarks from the **OpenCV library** as the model will be using these to determine which sign is present in the frame. This took approximately **100+ hours of training time**.

A **MobileNet** model pre-trained on ImageNet is used for **image classification** and fined tuned to a dataset of the ASL alphabet and all the training was done in Google Co-Lab



### Web Application





#### Main Page

When visiting the main page, the user will perform the current sign and when the start session button is pressed, it will capture your hand from the window.

#### Scores Page


On the scores page all the users scores for each letter are accessed along with progress bars, which can be filtered by highest score or by individual letter.

#### Register Page

Here is where the users will login and register. All passwords are encrypted, and the user cannot access the main page features when not logged in.

### Results

This is the output from the model being run through a python script, which sends the frames from the local camera through the model and displays **real-time predictions** on the screen.



The confusion matrix shows how well the model classifies each ASL letter along the diagonal. The figure on the right is testing inference which is the process of using a trained model to make predictions or decisions based on new, unseen data. The model has **~2.6 million trainable parameters** and **37,184 non-trainable parameters**.

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
functional_11 (Functional)	(None, 26)	2,598,746

Total params: 2,598,746 (9.91 MB)  
Trainable params: 2,561,562 (9.77 MB)  
Non-trainable params: 37,184 (145.25 KB)

