# PROJECT REPORT

DIVYANSH JAIN
MUDIT AGRAWAL
EBI JOHN SINJIN
KANISHK PRASAD

## Introduction

***Project Overview:*** The ASL Alphabet Image Recognition project aims to leverage machine learning to bridge the communication gap between the deaf and hearing communities by recognizing and translating American Sign Language (ASL) gestures. The primary goal is to develop a user-friendly website where individuals can upload images of ASL signs, and an AI model will accurately identify and translate the corresponding signs.

***Purpose:*** The purpose of this project is to enhance communication accessibility for the deaf community, providing a tool that can interpret ASL signs captured in images. By harnessing machine learning algorithms, we intend to create a reliable and efficient system that promotes inclusivity and facilitates better understanding between deaf and hearing individuals.

## Literature Survey

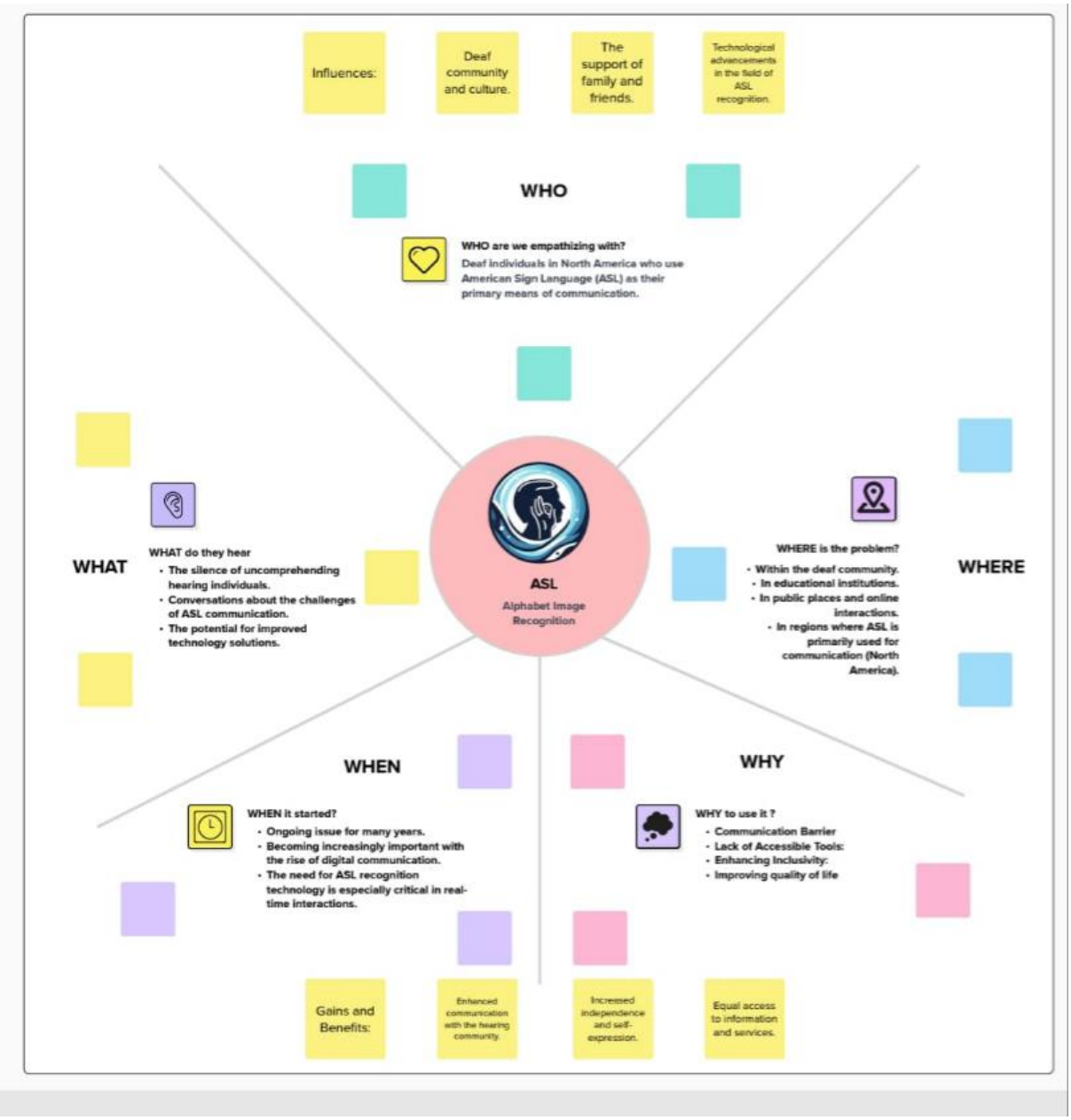
***Existing Problems:*** Numerous challenges exist in developing effective ASL image recognition systems, including variations in signing styles, lighting conditions, and background clutter. The literature review explores current technologies, their limitations, and potential areas for improvement.

***References:*** Key references in the field of ASL image recognition and related technologies are surveyed, including academic papers, research articles, and existing projects. Understanding the existing body of knowledge is crucial for building upon prior work and avoiding redundant efforts.

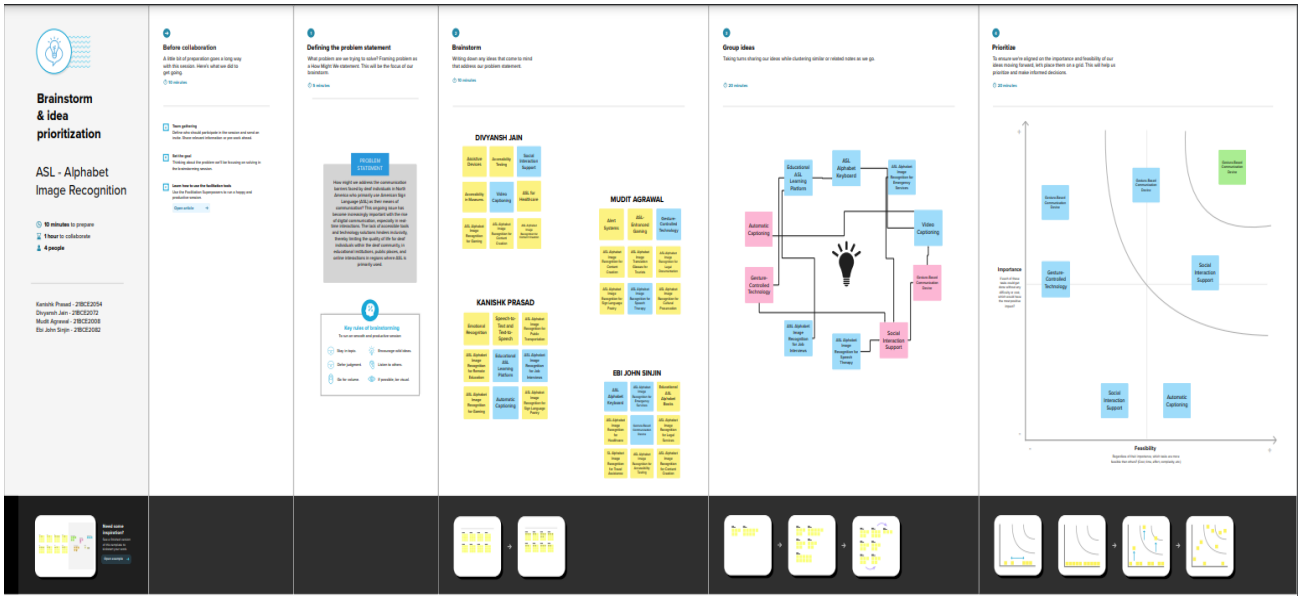
***Problem Statement Definitions:*** The problem statement outlines the specific challenges addressed by this project, such as accurate recognition of ASL signs, handling variations in gestures, and ensuring real-time processing for practical communication scenarios.

## Ideation & Proposed Solution

### Empathy Map Canvas:



## Ideation and Brainstorming Phase




**<u>These all are properly uploaded individually in Git Hub Repo</u>**

## Requirement Analysis

### *Functional Requirements:*

1. Image Upload: Users should be able to upload images containing ASL signs.
2. Image Recognition: The system must accurately identify and classify ASL signs.
3. Translation: A translation feature should convert recognized signs into English text.
4. User Authentication: Secure user authentication mechanisms to ensure data privacy.
5. Database Management: Efficient storage and retrieval of user-uploaded images and corresponding translations.

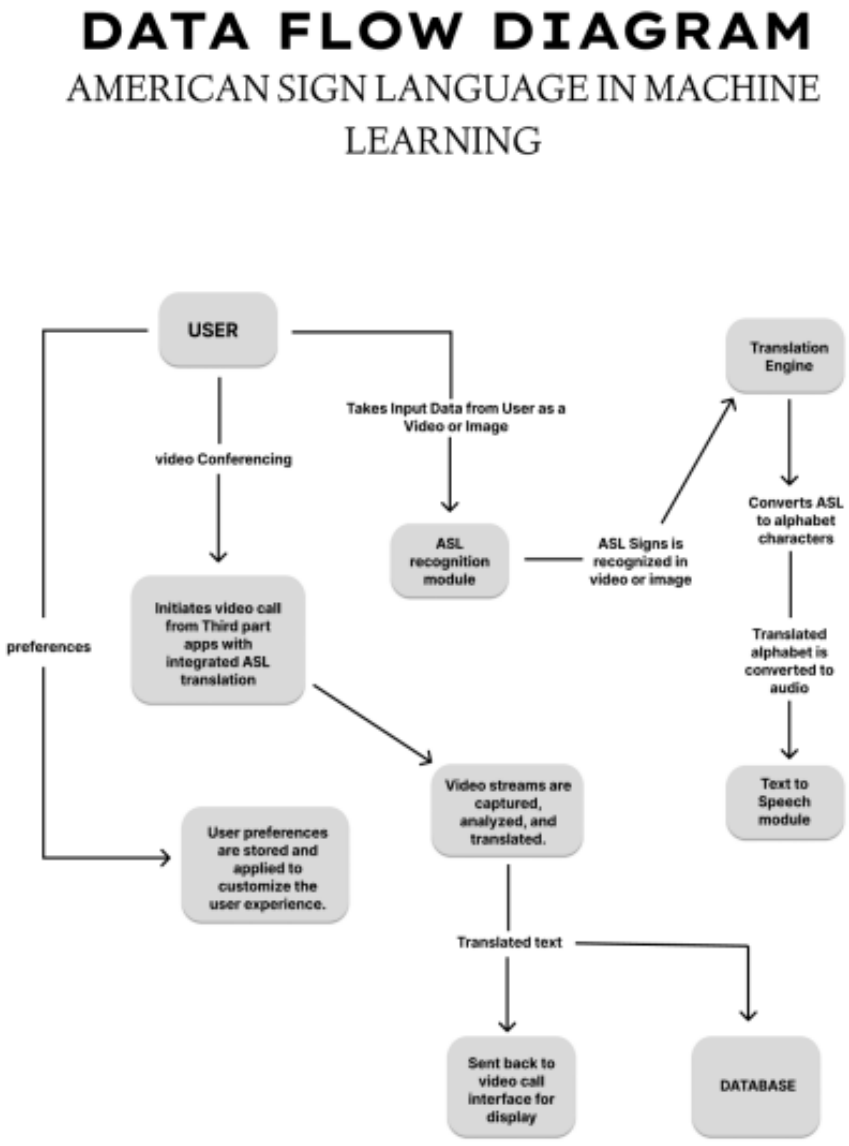
### *Non-Functional Requirements:*

1. Performance: Real-time processing for quick recognition.

2. Security: Robust security measures for user data protection.
3. Scalability: Ability to handle increasing user loads.
4. User Interface: Intuitive and user-friendly design for seamless interaction.
5. Compatibility: Support for various devices and browsers.

## Project Design

Data Flow Diagram:



Users Stories

# USER STORIES

| User Story Number | User Role | User Story | Status | Priority |
|---|---|---|---|---|
| US001 | Deaf User | As a Deaf user, I want to communicate using ASL in a video call, so others can understand me. | In Progress | High |
| US002 | Hearing User | As a Hearing user, I want to have the ASL signs translated to text during a video call, so I can understand Deaf users. | To-Do | High |
| US003 | Content Creator | As a Content Creator, I want to upload ASL videos, so the application can translate the signs and generate subtitles. | To-Do | Medium |
| US004 | Language Teacher | As a Language Teacher, I want to use the app for teaching ASL and have the option to adjust translation accuracy for my students. | To-Do | Medium |
| US005 | Admin/Support | As an Admin/Support user, I want to manage user accounts, ensure data security, and provide technical | To-Do | High |

## Project Planning and Scheduling

***Technical Architecture:*** The system will incorporate a convolutional neural network (CNN) for image classification, a web interface for user interaction, and a secure backend for data management. The architecture ensures efficient communication between these components.

***Sprint Planning and Estimation:*** Sprints will focus on specific functionalities, such as image recognition, translation, and user authentication. Each sprint will have defined goals and timelines, ensuring incremental progress.

***Sprint Delivery Schedule:*** Sprints will be delivered in regular intervals, with a clear timeline for each phase of development. This iterative approach allows for continuous improvement based on user feedback.

## Coding and Solutioning

https://github.com/smartinternz02/SI-GuidedProject-599493-1697481610/blob/main/Brainstroming%20Map%20ASL.pdf

Every required code are available here

## Performance Testing

1. Recognition Accuracy: Percentage of correctly identified ASL signs.
2. Response Time: Time taken for the system to process and translate an uploaded image.
3. Scalability: System's ability to handle increased user loads without compromising performance.

```
[ ] metrics = pd.DataFrame(model.history.history)
    print("The model metrics are")
    metrics
```

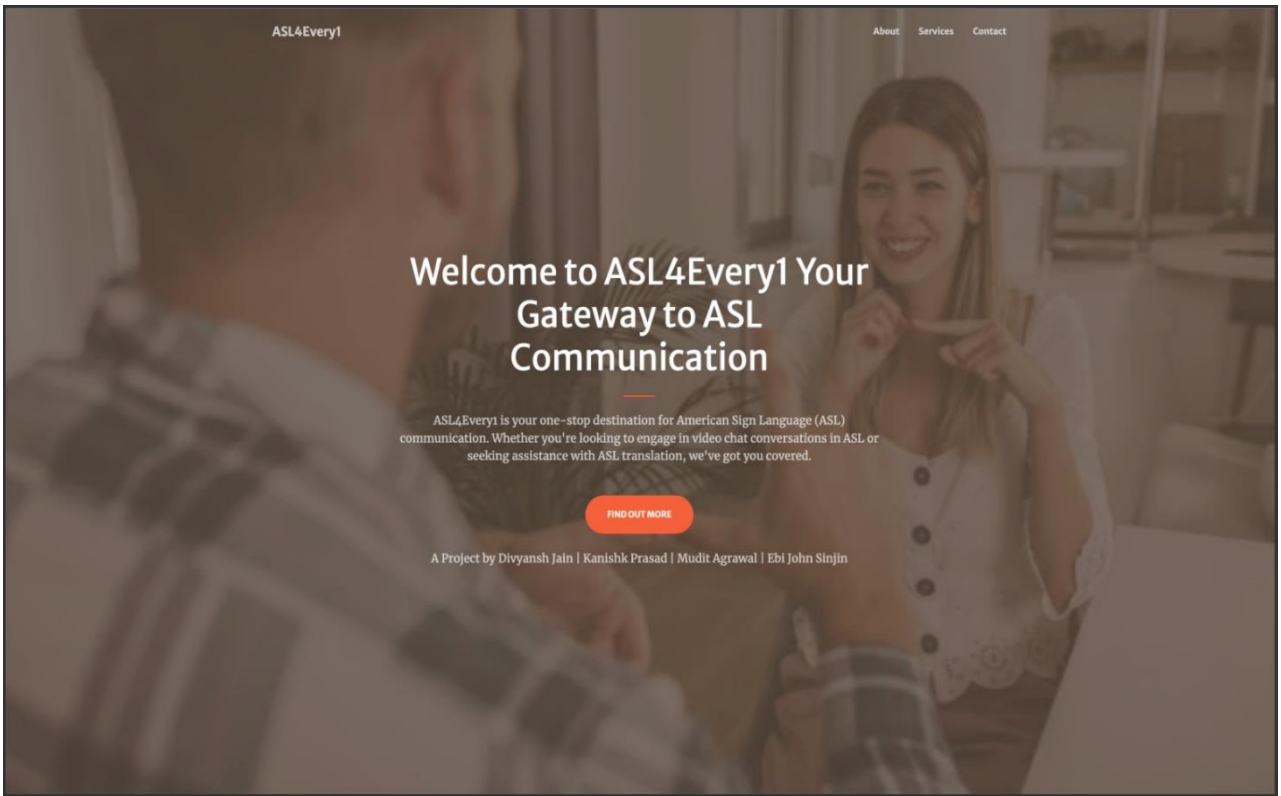
The model metrics are

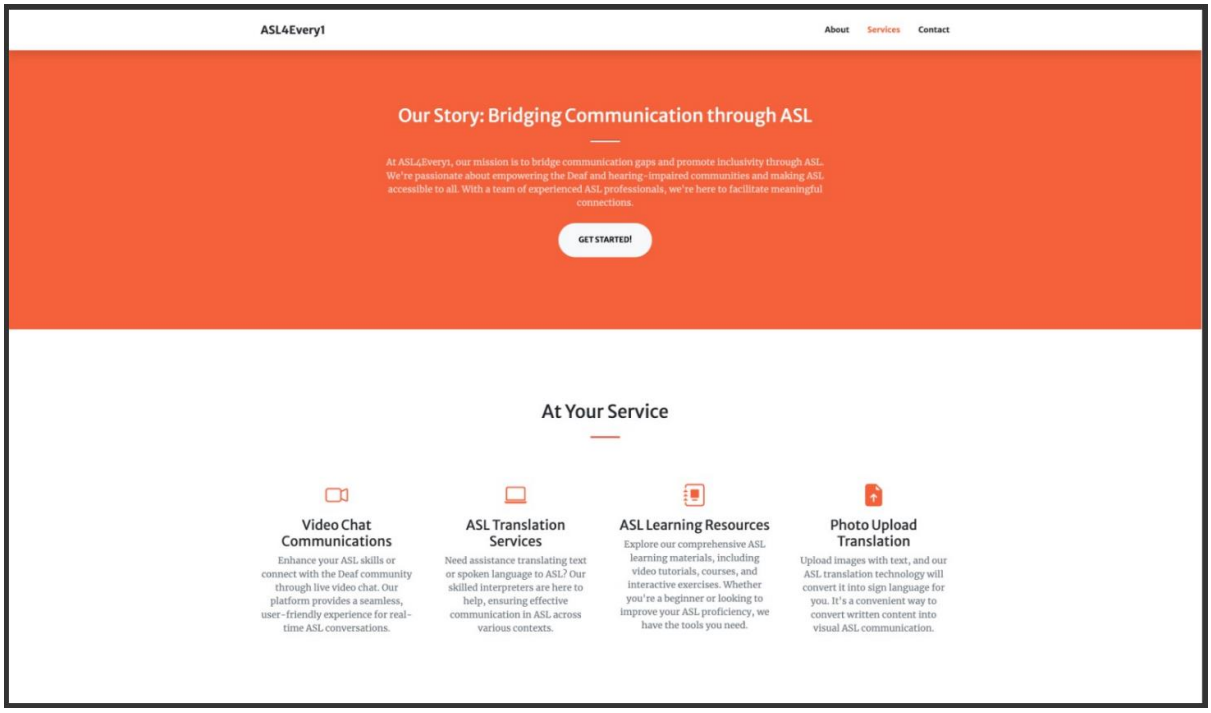
| | loss | accuracy | val_loss | val_accuracy |
|---|---|---|---|---|
| 0 | 1.093370 | 0.672299 | 0.286788 | 0.909157 |
| 1 | 0.158349 | 0.948424 | 0.106573 | 0.965900 |
| 2 | 0.078258 | 0.975172 | 0.193878 | 0.947663 |
| 3 | 0.049385 | 0.984581 | 0.053560 | 0.982988 |
| 4 | 0.044546 | 0.985665 | 0.031719 | 0.988391 |
| 5 | 0.030379 | 0.990788 | 0.040670 | 0.987433 |
| 6 | 0.037448 | 0.988752 | 0.047627 | 0.983716 |

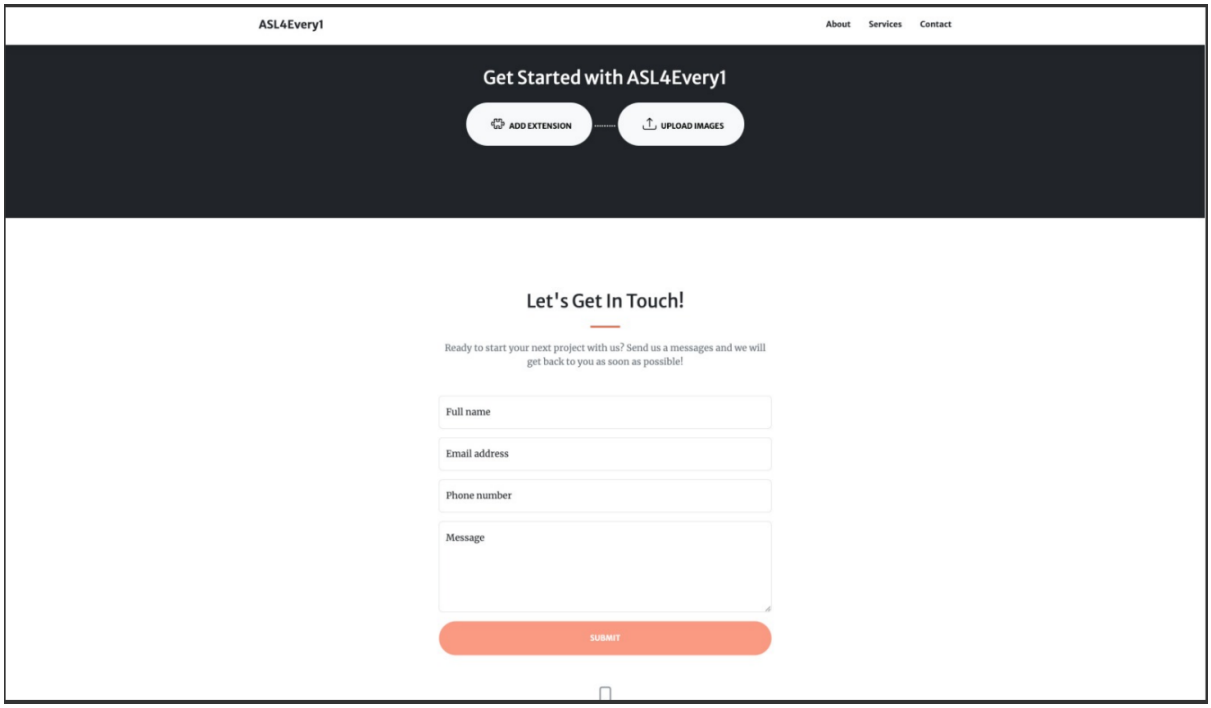
## Results

## OUTPUT SCREENSHOTS

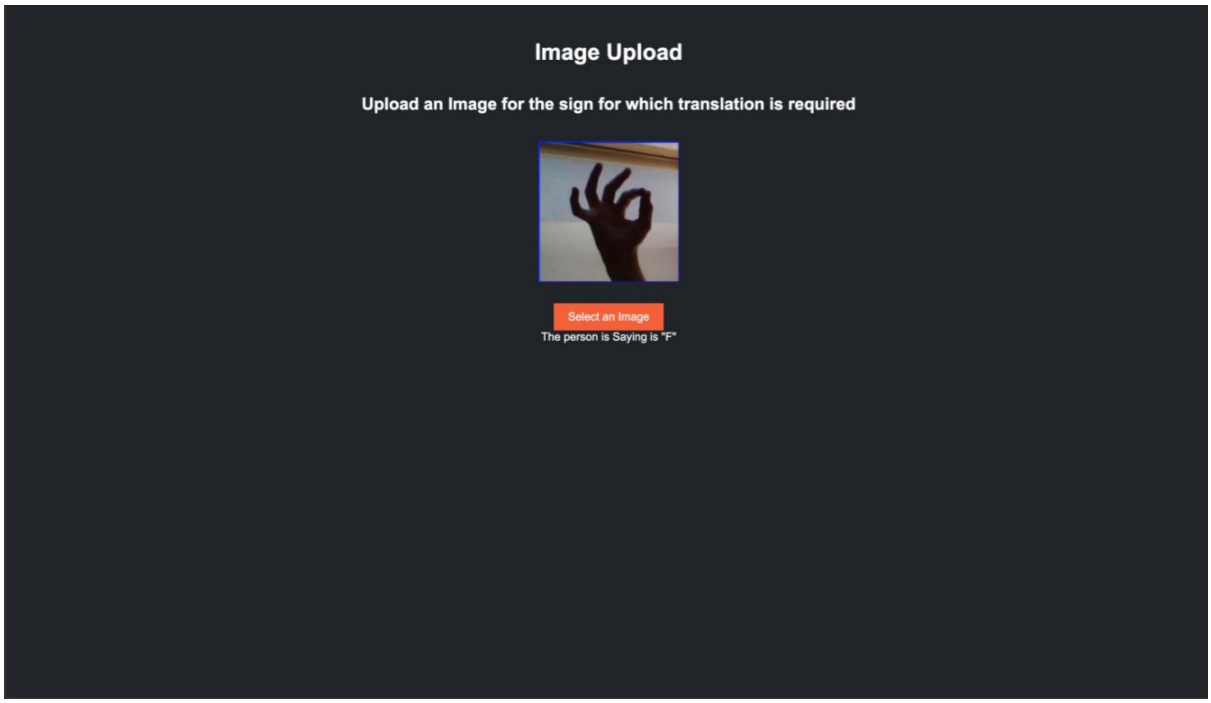
Below mentioned is our website for ASL recognition made by our team and have all required functionalities

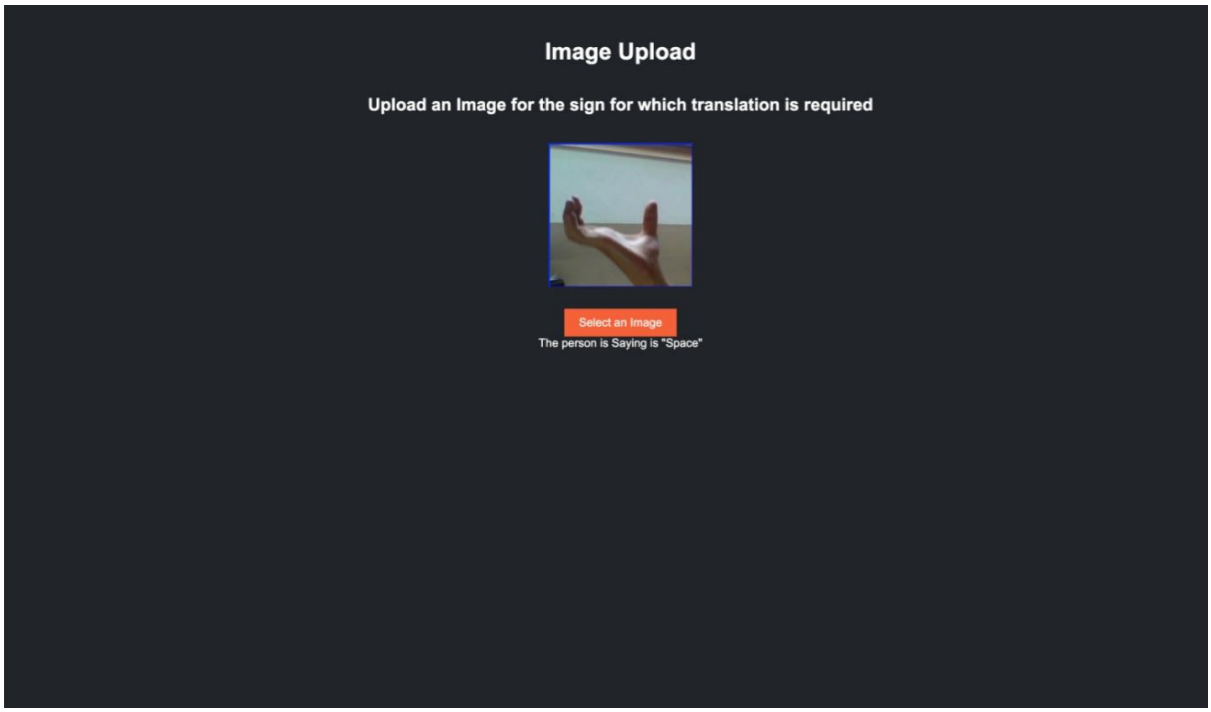



ASL4Every1
About
Services
Contact
Our Story: Bridging Communication through ASL
At ASL4Every1, our mission is to bridge communication gaps and promote inclusivity through ASL. We're passionate about empowering the Deaf and hearing-impaired communities and making ASL accessible to all. With a team of experienced ASL professionals, we're here to facilitate meaningful connections.
GET STARTED!
At Your Service
Video Chat Communications
Enhance your ASL skills or connect with the Deaf community through live video chat. Our platform provides a seamless, user-friendly experience for real-time ASL conversations.
ASL Translation Services
Need assistance translating text or spoken language to ASL? Our skilled interpreters are here to help, ensuring effective communication in ASL across various contexts.
ASL Learning Resources
Explore our comprehensive ASL learning materials, including video tutorials, courses, and interactive exercises. Whether you're a beginner or looking to improve your ASL proficiency, we have the tools you need.
Photo Upload Translation
Upload images with text, and our ASL translation technology will convert it into sign language for you. It's a convenient way to convert written content into visual ASL communication.


ASL4Every1
About
Services
Contact
Get Started with ASL4Every1
ADD EXTENSION
UPLOAD IMAGES
Let's Get In Touch!
Ready to start your next project with us? Send us a messages and we will get back to you as soon as possible!
Full name
Email address
Phone number
Message
SUBMIT

After doing all the required tasks and now we can see the main working as we upload the images our model recognizes the signs as shown below



## Image Upload

**Upload an Image for the sign for which translation is required**

Select an Image

The person is Saying is "K"

## Image Upload

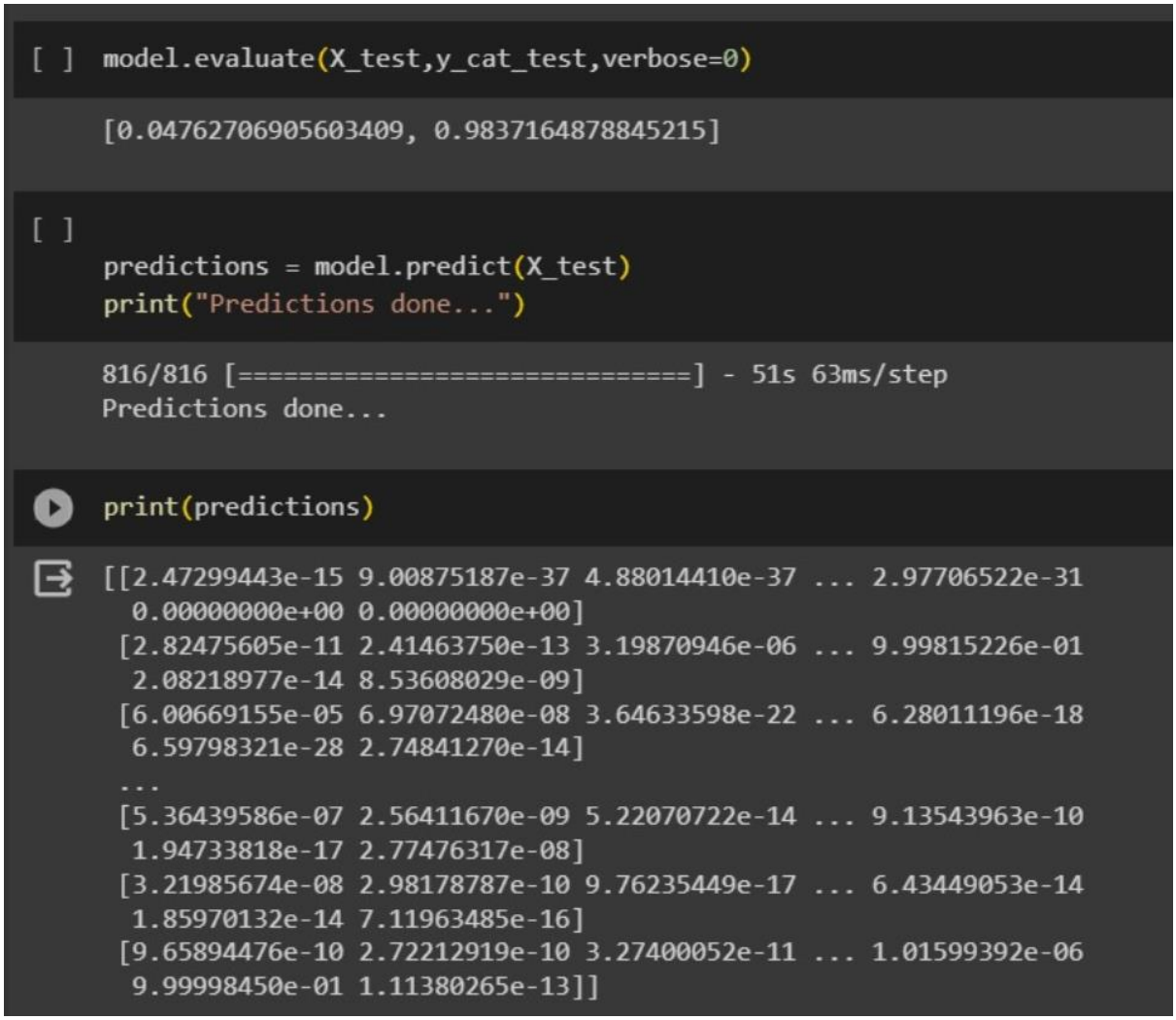
**Upload an Image for the sign for which translation is required**

Select an Image

The person is Saying is "Delete"

This would be the prediction on the model

All rest important codes, flask components are uploaded in GITHUB



```
[ ]  model.evaluate(X_test,y_cat_test,verbose=0)

     [0.04762706905603409, 0.9837164878845215]

[ ]
     predictions = model.predict(X_test)
     print("Predictions done...")

     816/816 [==============================] - 51s 63ms/step
     Predictions done...

     print(predictions)

     [[2.47299443e-15 9.00875187e-37 4.88014410e-37 ... 2.97706522e-31
       0.00000000e+00 0.00000000e+00]
      [2.82475605e-11 2.41463750e-13 3.19870946e-06 ... 9.99815226e-01
       2.08218977e-14 8.53608029e-09]
      [6.00669155e-05 6.97072480e-08 3.64633598e-22 ... 6.28011196e-18
       6.59798321e-28 2.74841270e-14]
      ...
      [5.36439586e-07 2.56411670e-09 5.22070722e-14 ... 9.13543963e-10
       1.94733818e-17 2.77476317e-08]
      [3.21985674e-08 2.98178787e-10 9.76235449e-17 ... 6.43449053e-14
       1.85970132e-14 7.11963485e-16]
      [9.65894476e-10 2.72212919e-10 3.27400052e-11 ... 1.01599392e-06
       9.99998450e-01 1.11380265e-13]]
```

To launch the Flask application, simply follow these instructions:

1. Go to the project directory using your terminal or command prompt.

2. Run the command "python app.py" in the terminal to initiate the Flask server.

3. After starting the server, you can access the web application by opening your web browser and navigating to the local host address.

4. Once the application loads, you can upload an image and submit the form.

5. The application will then utilize the trained model to predict the plant species in the image.

6. The prediction outcome will be presented on the web page.

By adhering to these steps, you can interact with the Flask application, upload images, and obtain plant species predictions.

## Advantages and Disadvantages

***Advantages:***

1. Enhanced communication accessibility for the deaf community.
2. Improved understanding between deaf and hearing individuals.
3. User-friendly interface for easy interaction.

***Disadvantages:***

1. Dependence on image quality and clarity.
2. Potential challenges in handling diverse signing styles.

## Conclusions

The ASL Alphabet Image Recognition project represents a significant step towards leveraging technology for fostering communication inclusivity. Despite challenges, the system's capabilities provide a foundation for future improvements and expansions.

## Future Scope

Future developments may include:

1. Integration of real-time video recognition for dynamic sign language communication.
2. Continuous training of the model to adapt to evolving signing styles.

3. Collaboration with mobile applications for on-the-go accessibility.

## Appendix

## Source code and Project Demo link are Uploaded in Git Repo Collaboration