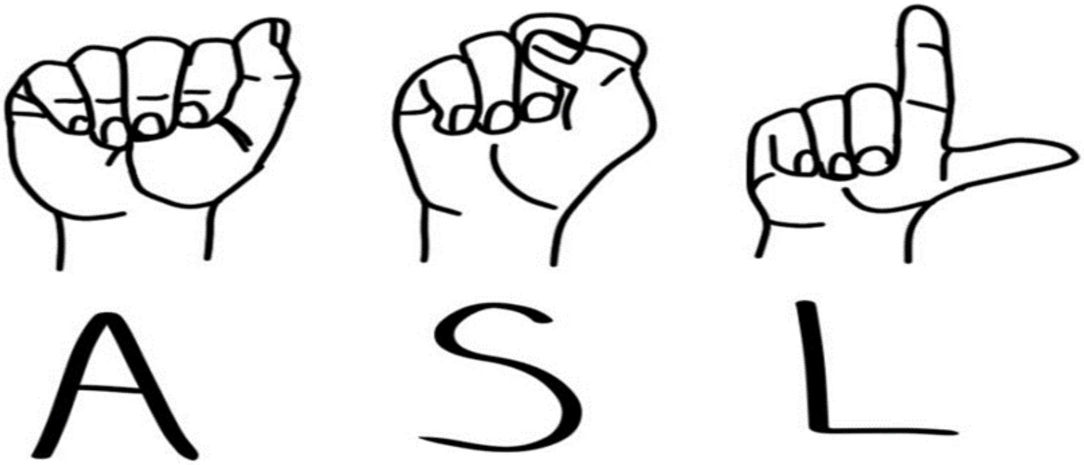
ASL- Alphabet Image Recognition



Project Title : ASL Alphabet Image Recognition

Duration : 30 days

Mentor Name : Saumya Mohandas

TEAM: **Shreyas Johri (Shreyas.21bcb7015@vitapstudent.ac.in)**

Amajala Sairam(sai.21bce8231@vitapstudent.ac.in)

Haneesh Sai(haneesh.21bcb7034@vitapstudent.ac.in)

Chandreka mutukumdu(chandreka.21bcb7009@vitapstudent.ac.in)

1.INTRODUCTION

1.1 Project Overview :

Objectives:

1. Train an AI model to accurately classify ASL alphabet hand signs.

2. Extend the model's capabilities to recognize special signs for "space," "delete," and "nothing."

3. Develop a deployable model suitable for integration into real-time applications, allowing for instantaneous ASL interpretation.

Motivation:

The project is motivated by the need to address communication challenges faced by individuals using ASL. By utilizing image recognition technology, we aim to create a tool that enhances accessibility and promotes inclusivity by enabling effective communication between individuals with varying linguistic abilities.

Scope:

The project's scope includes the development of a specialized image recognition model focused on ASL hand signs. Potential applications include real-time interpretation tools, educational resources, and communication aids. The project emphasizes the intersection of technology and accessibility to foster a more connected and inclusive society. In subsequent sections, we will detail the methodology, data collection and preparation, model development, training, evaluation metrics, and deployment strategy for the ASL Alphabet Image Recognition project.

1.2 Purpose:

- Bridging the Communication Gap with ASL Alphabet Image Recognition
- Addressing the Communication Divide
- Enhancing Accessibility and Inclusivity

2.LITERATURE SURVEY

2.1 Existing problem :

- Insufficient Familiarity with ASL

  ASL, being a language that relies on visual and gestural communication, is not universally comprehended, resulting in misunderstandings and impeding effective communication between deaf and hearing individuals. In situations where ASL is unfamiliar, the deaf community experiences isolation and struggles to express their thoughts, ideas, and requirements.

- Challenges in Educational and Workplace Environments

In educational and workplace environments, the absence of ASL proficiency among classmates and colleagues can hinder the complete involvement and inclusion of deaf individuals. This can lead to missed chances for collaboration and impede their educational and professional progress.

In day-to-day encounters, basic activities such as placing a food order, asking for directions, or having casual conversations can pose difficulties for individuals who rely on ASL. The lack of a universally accepted and comprehensible means of interpreting ASL often results in feelings of frustration and being left out.

2.2 References

Encyclopædia Britannica, inc. (2023, November 15). *American sign language*. Encyclopædia Britannica. https://www.britannica.com/topic/American-Sign-Language

Yasar, K. (2023, March 15). *What is image recognition?: Definition from TechTarget*. Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/image-recognition

Image recognition: Definition, algorithms & uses. V7. (n.d.). https://www.v7labs.com/blog/image-recognition-guide

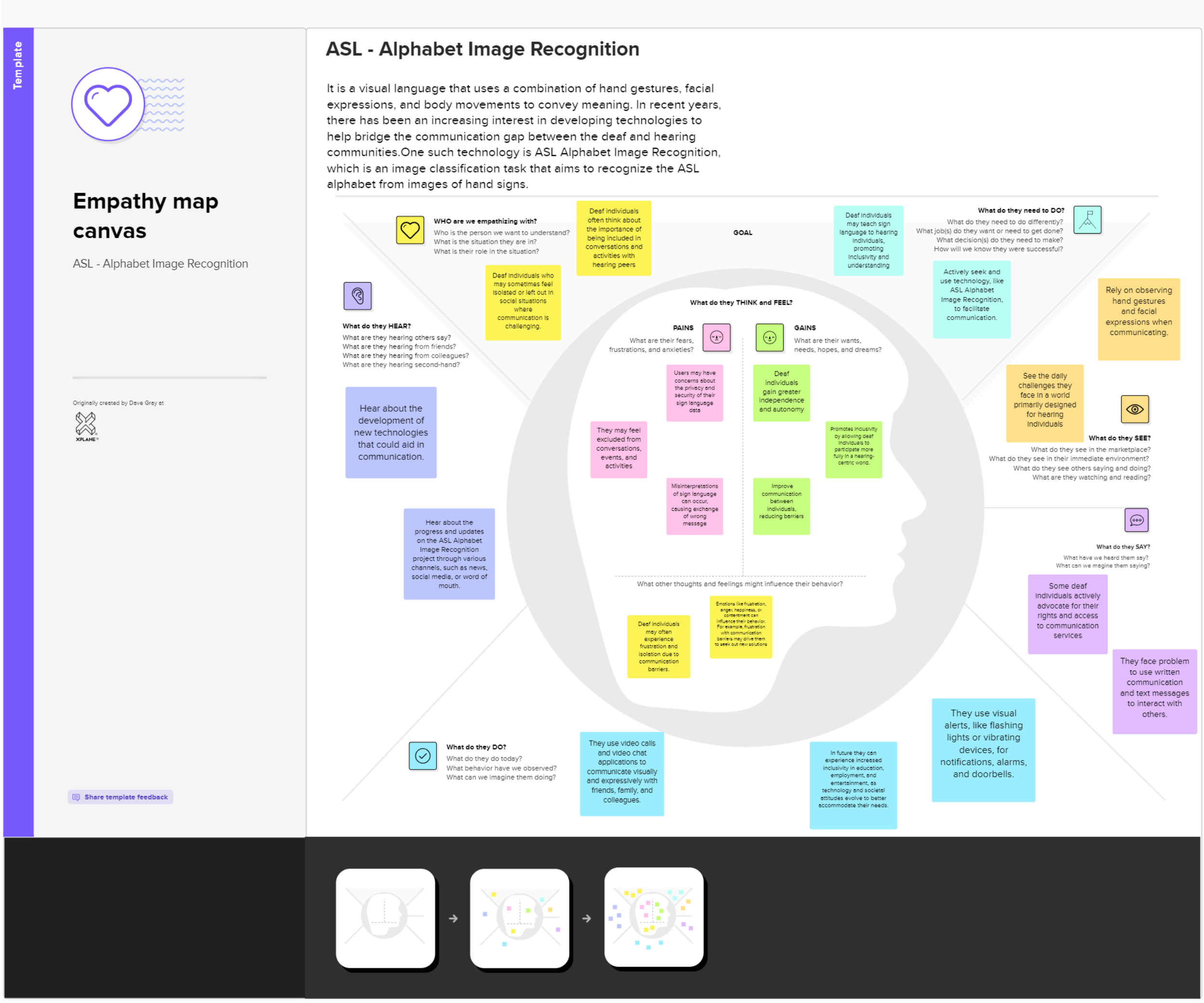
2.3 Problem Statement Definition:

Deaf individuals face challenges in effectively communicating, which results in feelings of isolation, limited educational and professional opportunities, and difficulties in everyday interactions. This problem is exacerbated by the absence of a widely recognized method for interpreting ASL hand signs. The objective of this project is to tackle this issue by utilizing machine learning and image recognition technologies to create a system that can interpret the ASL alphabet in real-time, including special signs for "space," "delete," and "nothing." The ultimate aim is to bridge communication gaps, empower the deaf community, and foster inclusivity in various social contexts.

3.IDEATION & PROPOSED SOLUTION

Empathy map canvas, which helped to understand the needs, challenges, and emotions of deaf individuals who use ASL as their primary mode of communication. The project also aims to promote inclusivity and understanding of the deaf community and their culture.

3.1 Empathy Map Canvas :



Empathy map canvas

ASL - Alphabet Image Recognition

Originally created by Dave Gray at

ASL - Alphabet Image Recognition

It is a visual language that uses a combination of hand gestures, facial expressions, and body movements to convey meaning. In recent years, there has been an increasing interest in developing technologies to help bridge the communication gap between the deaf and hearing communities.One such technology is ASL Alphabet Image Recognition, which is an image classification task that aims to recognize the ASL alphabet from images of hand signs.

WHO are we empathizing with?

Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

Deaf individuals often think about the importance of being included in conversations and activities with hearing peers

Deaf individuals who may sometimes feel isolated or left out in social situations where communication is challenging.

GOAL

Deaf individuals may teach sign language to hearing individuals, promoting inclusivity and understanding

What do they need to DO?

What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

Actively seek and use technology, like ASL Alphabet Image Recognition, to facilitate communication.

What do they HEAR?

What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

Hear about the development of new technologies that could aid in communication.

Hear about the progress and updates on the ASL Alphabet Image Recognition project through various channels, such as news, social media, or word of mouth.

What do they THINK and FEEL?

PAINS

What are their fears, frustrations, and anxieties?

Users may have concerns about the privacy and security of their sign language data

They may feel excluded from conversations, events, and activities

Misinterpretations of sign language can occur, causing exchange of wrong message

GAINS

What are their wants, needs, hopes, and dreams?

Deaf individuals gain greater independence and autonomy

Promotes inclusivity by allowing deaf individuals to participate more fully in a hearing-centric world.

Improve communication between individuals, reducing barriers

What other thoughts and feelings might influence their behavior?

Deaf individuals may often experience frustration and isolation due to communication barriers.

Emotions like frustration, anger, happiness, or contentment can influence their behavior. For example, frustration with communication barriers may drive them to seek out new solutions

What do they SEE?

What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

Rely on observing hand gestures and facial expressions when communicating.

See the daily challenges they face in a world primarily designed for hearing individuals

What do they SAY?

What have we heard them say?
What can we magine them saying?

Some deaf individuals actively advocate for their rights and access to communication services

They face problem to use written communication and text messages to interact with others.

What do they DO?

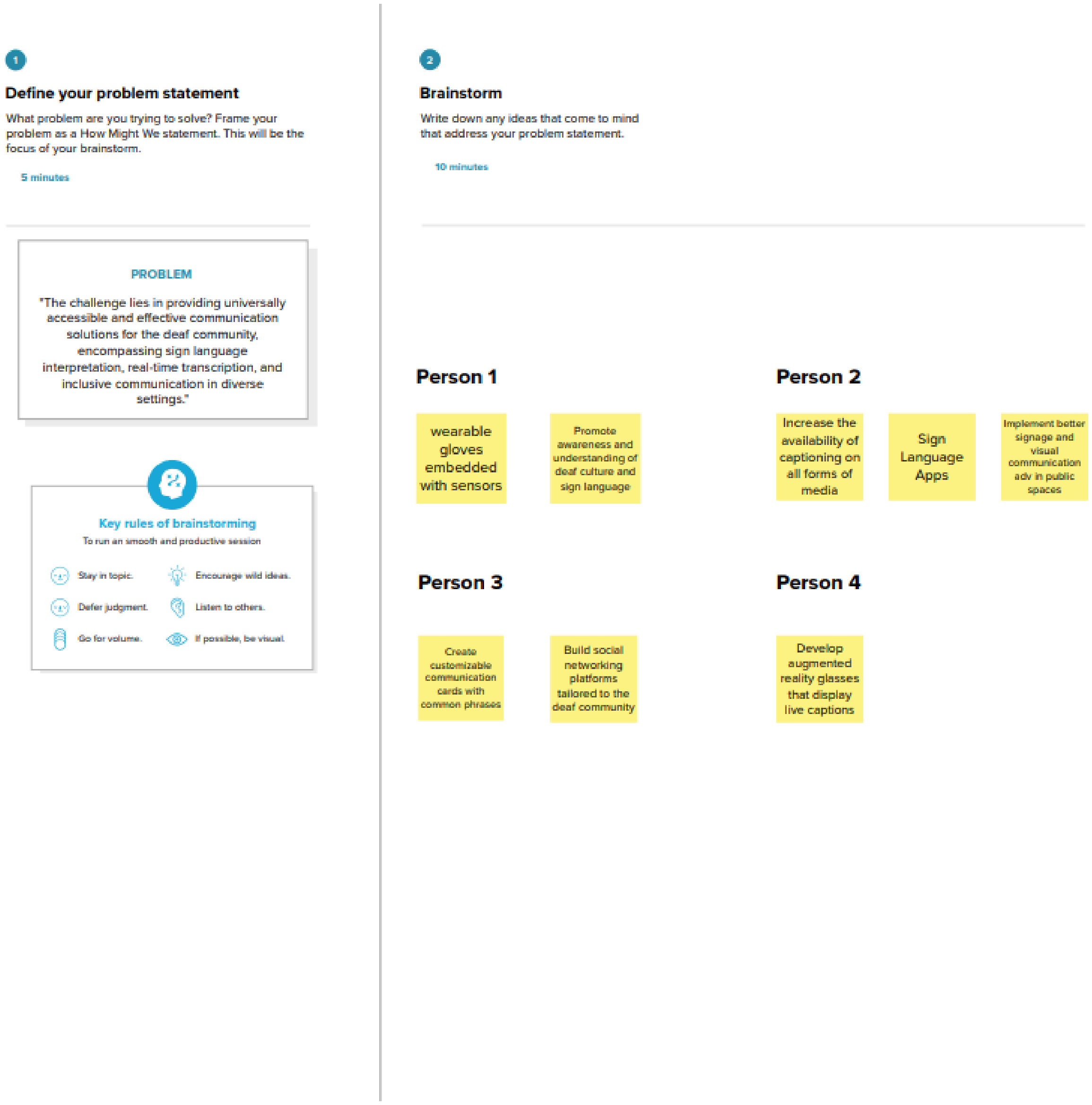
What do they do today?
What behavior have we observed?
What can we imagine them doing?

They use video calls and video chat applications to communicate visually and expressively with friends, family, and colleagues.

In future they can experience increased inclusivity in education, employment, and entertainment, as technology and societal attitudes evolve to better accommodate their needs.

They use visual alerts, like flashing lights or vibrating devices, for notifications, alarms, and doorbells.

3.2 Ideation & Brainstorming :



1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

"The challenge lies in providing universally accessible and effective communication solutions for the deaf community, encompassing sign language interpretation, real-time transcription, and inclusive communication in diverse settings."

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2 **Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

Person 1

- wearable gloves embedded with sensors
- Promote awareness and understanding of deaf culture and sign language

Person 2

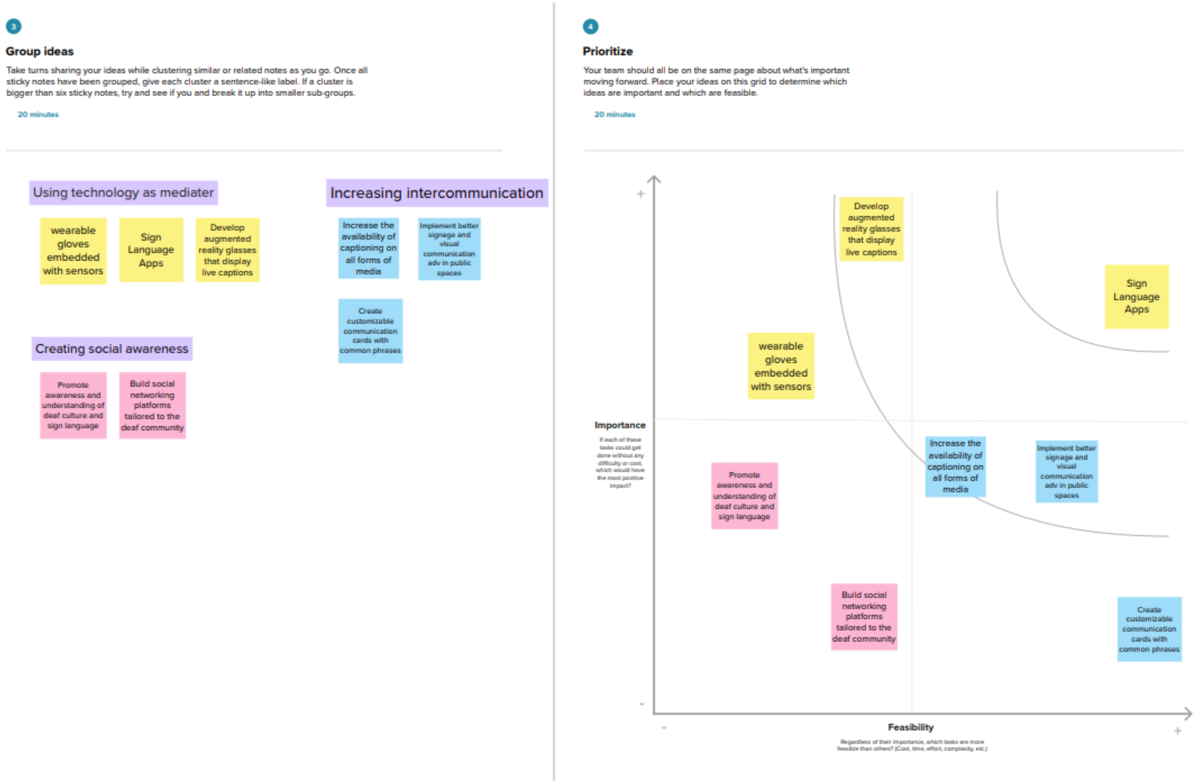
- Increase the availability of captioning on all forms of media
- Sign Language Apps
- Implement better signage and visual communication adv in public spaces

Person 3

- Create customizable communication cards with common phrases
- Build social networking platforms tailored to the deaf community

Person 4

- Develop augmented reality glasses that display live captions



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Using technology as mediater
wearable gloves embedded with sensors
Sign Language Apps
Develop augmented reality glasses that display live captions
Increasing intercommunication
Increase the availability of captioning on all forms of media
Implement better signage and visual communication adv in public spaces
Create customizable communication cards with common phrases
Creating social awareness
Promote awareness and understanding of deaf culture and sign language
Build social networking platforms tailored to the deaf community
+
Develop augmented reality glasses that display live captions
Sign Language Apps
wearable gloves embedded with sensors
Increase the availability of captioning on all forms of media
Implement better signage and visual communication adv in public spaces
Promote awareness and understanding of deaf culture and sign language
Build social networking platforms tailored to the deaf community
Create customizable communication cards with common phrases
-
-
+

4.REQUIREMENT ANALYSIS

4.1 Functional requirement :

- Image Classification

ASL Alphabet Recognition: The system needs to accurately categorize hand gestures that represent the 26 letters of the English alphabet in American Sign Language (ASL).

Recognition of Special Signs: In addition to the alphabet, the system should be able to identify special signs for "space," "delete," and "nothing."

- Real-time Interpretation

Instantaneous Processing: The system must be capable of performing image recognition in real-time, enabling immediate interpretation of ASL hand gestures from video streams.

Minimal Delay: The system should maintain a low latency to ensure prompt and responsive interpretation during live interactions.

- Model Training and Adaptability

Training the Model: The system should allow for the retraining of the model using additional data to improve recognition accuracy over time.

- Adaptability to Variations

  The model must be able to recognize variations in hand gestures caused by factors such as different hand shapes and orientations.

- Intuitive Interface

  The interface of the application must be designed in a way that is easy to use and understand for the users.

- Feedback Provision

  The system should provide feedback to the users regarding the hand sign recognition, which will help them in comprehending the interpretation.

4.2 Non-Functional requirements :

- Performance:

  The system should provide real-time recognition of ASL signs, with a response time of less than 1 second for optimal user experience.

- Compatibility:

  The system should be compatible with popular web browsers (such as Chrome, Firefox, and Safari) and mobile devices (iOS and Android).

- Maintainability:

  Usability:The user interface should be intuitive and easy to navigate, ensuring that users with varying levels of technical expertise can use the system effectively.

- Accessibility:

  The web application aims to enhance accessibility for deaf individuals, providing a means of communication through sign language recognition.

- User friendly interface:

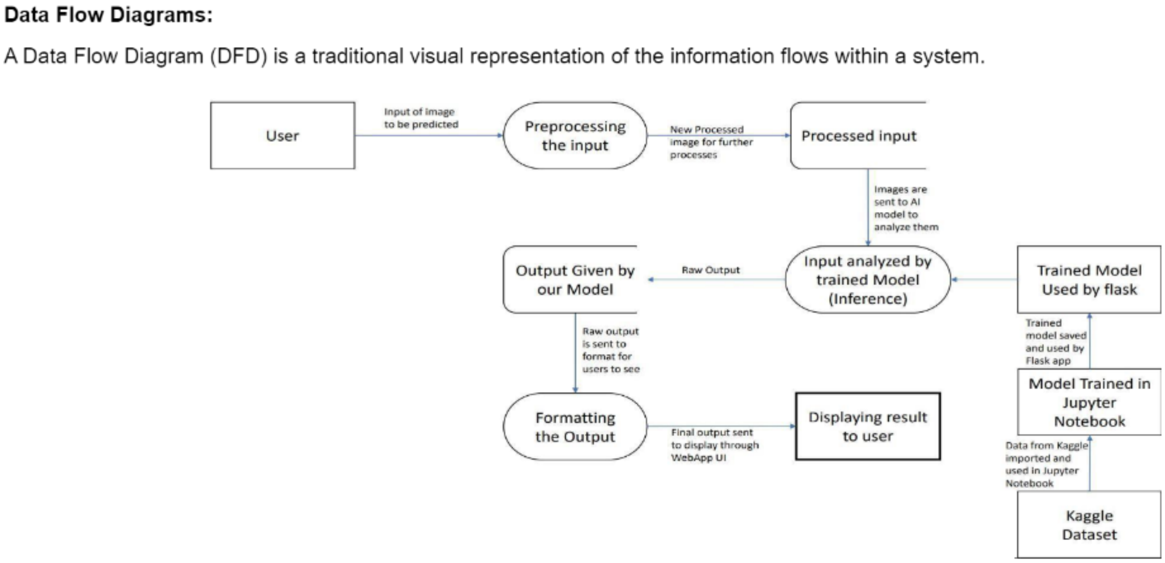
  The web application provides an intuitive and visually appealing user interface for users to interact with and easily submit hand sign images.

- Customization:

  The Python backend allows for easy customization and addition of features, ensuring flexibility in adapting to future requirements or improvements.

5.PROJECT DESIGN

5.1 Data Flow Diagrams (DFD)& User Stories :



Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system.

User
Input of image to be predicted
Preprocessing the input
New Processed image for further processes
Processed input
Images are sent to AI model to analyze them
Output Given by our Model
Raw Output
Input analyzed by trained Model (Inference)
Trained Model Used by flask
Raw output is sent to format for users to see
Trained model saved and used by Flask app
Model Trained in Jupyter Notebook
Formatting the Output
Final output sent to display through WebApp UI
Displaying result to user
Data from Kaggle imported and used in Jupyter Notebook
Kaggle Dataset

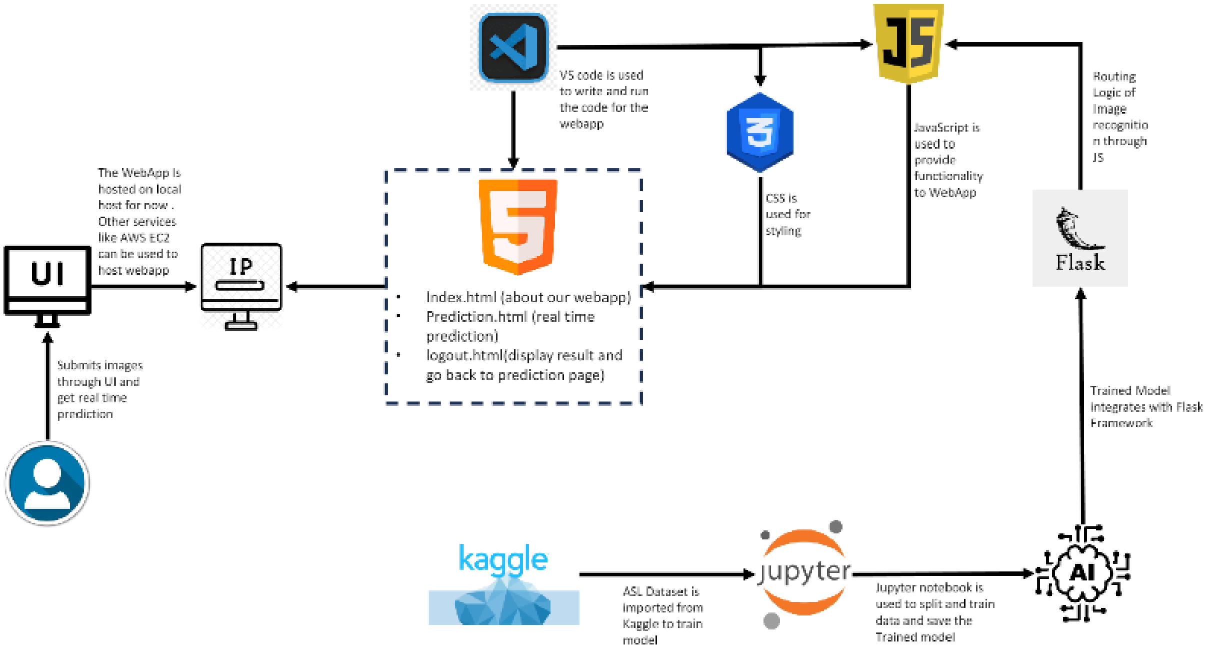
User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Annotation	USN-1	As a User, I want to see additional information about the predicted ASL hand sign, so I can learn more about it.	The prediction.html page should provide details such as the letter represented by the hand sign, as well as a brief description.	low	Sprint-4
Customer (Web user)	Gesture recognition	USN-2	As a User, I want the web application to recognize special gestures like "space," "delete," and "nothing."	The web application should be able to identify and appropriately handle special gestures, providing accurate predictions.	High	Sprint-2
Customer (Web user)	Web App Design	USN-3	I want to visit the web application and experience a user-friendly interface that allows me to seamlessly navigate through the ASL alphabet learning process.	The home page should have clear instructions and a welcoming design. Button for prediction for submitting hand signs should be intuitive. The webapp page should present results in a visually appealing format.	medium	Sprint-3
Customer (Web user)	Feedback mechanism	USN-4	As a User, I want the option to provide feedback on the prediction results, so I can contribute to improving the system.	The web application should include a way to contact to our team so user can express themselves	Medium	Sprint-2
Customer (Web user)	Real-Time ASL Alphabet Recognition	USN – 5	I want to submit images of ASL hand signs and receive real-time predictions to gauge the accuracy of my signs.	The web application should process and predict ASL hand signs in real-time.	High	Sprint-1
Customer (Web user)	Navigation	USN- 6	As a User, I want the option to navigate back to the home page after receiving the prediction, so I can submit additional images.	The result page should include a button or link that allows users to return to the home page for further interactions.	High	Sprint-3
Admin	Model Training and Maintenance	USN-7	I want to be able to retrain the machine learning model with new data to enhance its accuracy and ensure it stays up-to-date.	There should be a secure and efficient mechanism to update the training dataset.	Medium	Sprint-1
Admin	WebApp design	USN – 8	I want to change some designing in the logout page as for now it looks too simple.	Dev's should be able to easily change the design of webapp to make it more according to customers .	Low	Sprint -4

5.2 Solution Architecture :

Solution architecture follows a systematic approach that identifies your desired solution and the building blocks needed to construct it.


VS code is used to write and run the code for the webapp
CSS is used for styling
JavaScript is used to provide functionality to WebApp
Routing Logic of Image recognitio n through JS
Flask
The WebApp Is hosted on local host for now . Other services like AWS EC2 can be used to host webapp
UI
IP
Index.html (about our webapp)
Prediction.html (real time prediction)
logout.html(display result and go back to prediction page)
Submits images through UI and get real time prediction
Trained Model Integrates with Flask Framework
kaggle
ASL Dataset is imported from Kaggle to train model
jupyter
Jupyter notebook is used to split and train data and save the Trained model
AI

6.PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture & Stack :

Technology architecture deals with the deployment of application components on technology components.

Technical Architecture:

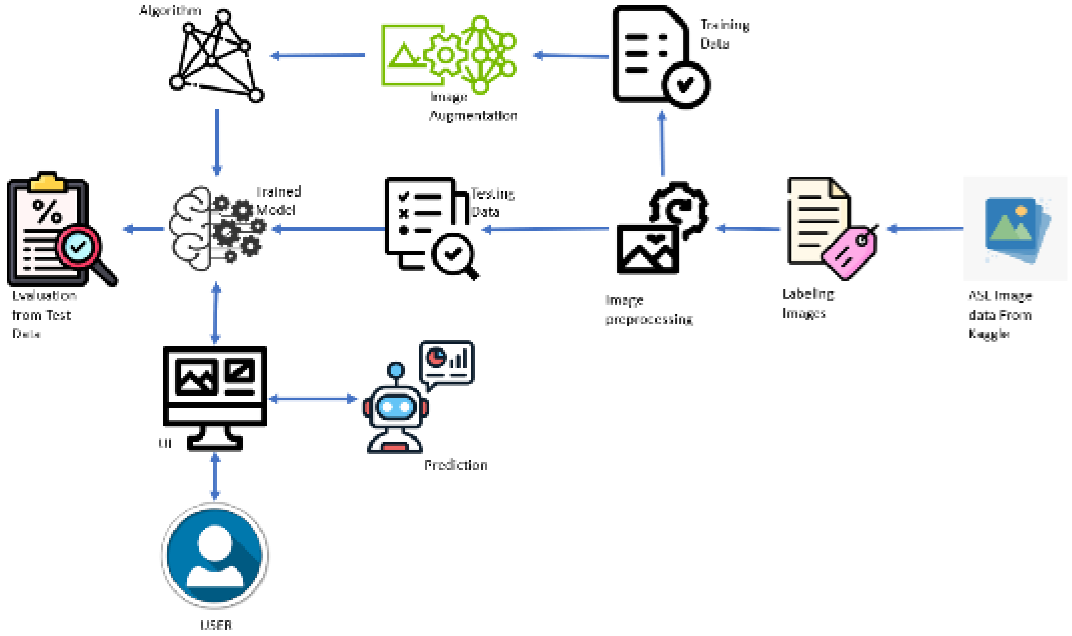


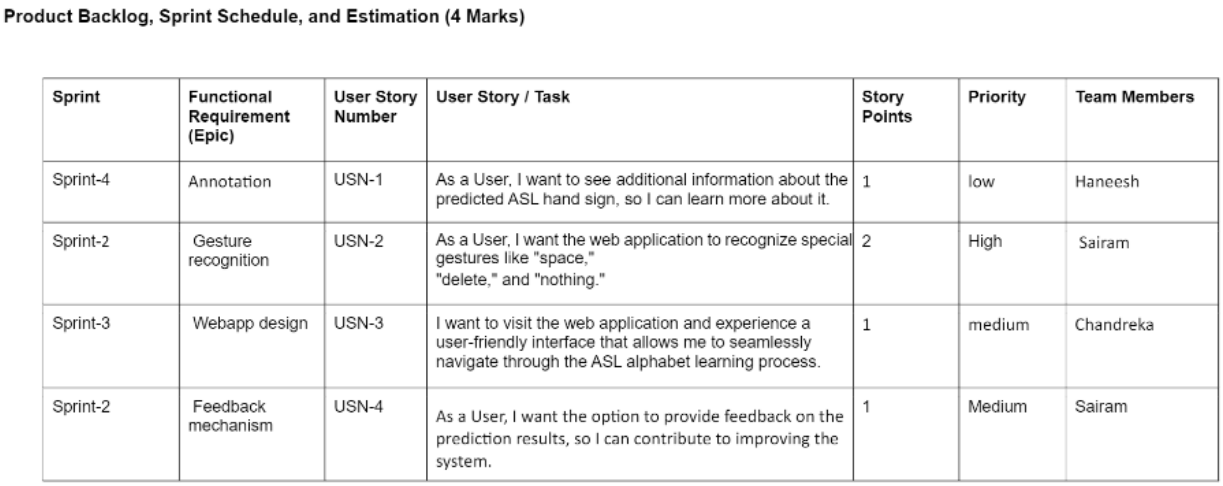
Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application	HTML, CSS, JavaScript
2.	Web Application Framework	Handles routing, request handling, and integrates the machine learning model.	Python Flask
3.	Machine Learning Model	Used for transfer learning in image classification.	VGG16
4.	Data Preprocessing	Used for data exploration, model training, and experimentation.	Jupyter Notebook
5.	Environment	Required for running Flask and the machine learning model.	Python Environment
6.	File Storage	File storage requirements	Local Filesystem
7.	Deployment Technologies	Integrated development environment for building and managing the Flask app.	VS Code
8.	Machine Learning Technologies	Used for building and training the machine learning model	TensorFlow Keras Libraries , sklearn and other libraries

Table-2: Application Characteristics:

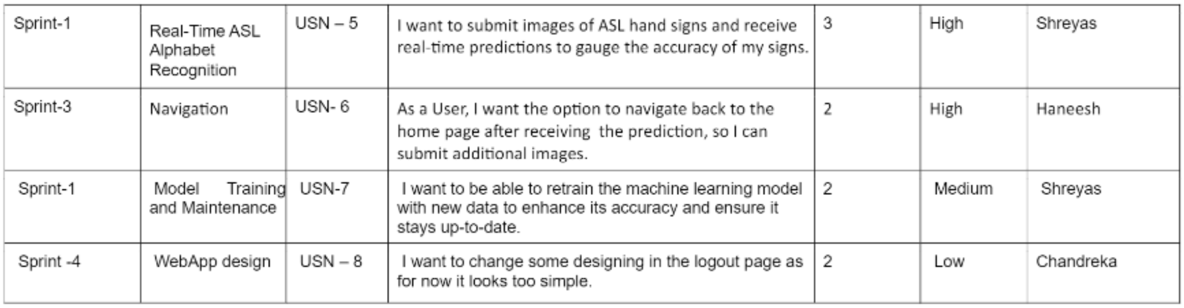
S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open-source frameworks used	Flask Python
2.	Accessibility	The web application aims to enhance accessibility for deaf individuals, providing a means of communication through sign language recognition.	ASL Alphabet Image Recognition Model
3.	User-Friendly Interface	The web application provides an intuitive and visually appealing user interface for users to interact with and easily submit hand sign images.	HTML5, CSS3, JavaScript
4.	Real-Time Prediction	The web application leverages Flask and Python to facilitate real-time processing and prediction of hand sign images using the trained machine learning model.	Flask Webapp
5.	Customization	The Python backend allows for easy customization and addition of features, ensuring flexibility in adapting to future requirements or improvements.	Python
6.	Ease of Deployment	Deployment is made easy using VS Code for development and ensuring efficient deployment and accessibility for users.	VS Code
7.	Cross-Browser Compatibility	The use of modern web technologies ensures cross-browser compatibility, allowing users to access the application from various web browsers.	HTML5, CSS3, JavaScript

6.2 Sprint Planning & Sprint Delivery :

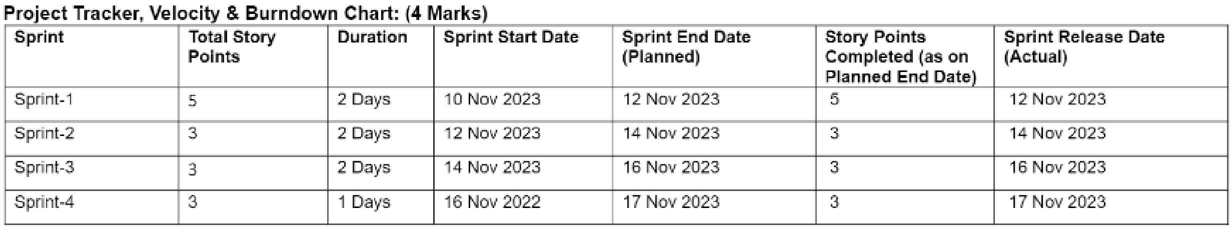


Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Annotation	USN-1	As a User, I want to see additional information about the predicted ASL hand sign, so I can learn more about it.	1	low	Haneesh
Sprint-2	Gesture recognition	USN-2	As a User, I want the web application to recognize special gestures like "space," "delete," and "nothing."	2	High	Sairam
Sprint-3	Webapp design	USN-3	I want to visit the web application and experience a user-friendly interface that allows me to seamlessly navigate through the ASL alphabet learning process.	1	medium	Chandreka
Sprint-2	Feedback mechanism	USN-4	As a User, I want the option to provide feedback on the prediction results, so I can contribute to improving the system.	1	Medium	Sairam



Sprint-1	Real-Time ASL Alphabet Recognition	USN – 5	I want to submit images of ASL hand signs and receive real-time predictions to gauge the accuracy of my signs.	3	High	Shreyas
Sprint-3	Navigation	USN- 6	As a User, I want the option to navigate back to the home page after receiving the prediction, so I can submit additional images.	2	High	Haneesh
Sprint-1	Model Training and Maintenance	USN-7	I want to be able to retrain the machine learning model with new data to enhance its accuracy and ensure it stays up-to-date.	2	Medium	Shreyas
Sprint -4	WebApp design	USN – 8	I want to change some designing in the logout page as for now it looks too simple.	2	Low	Chandreka

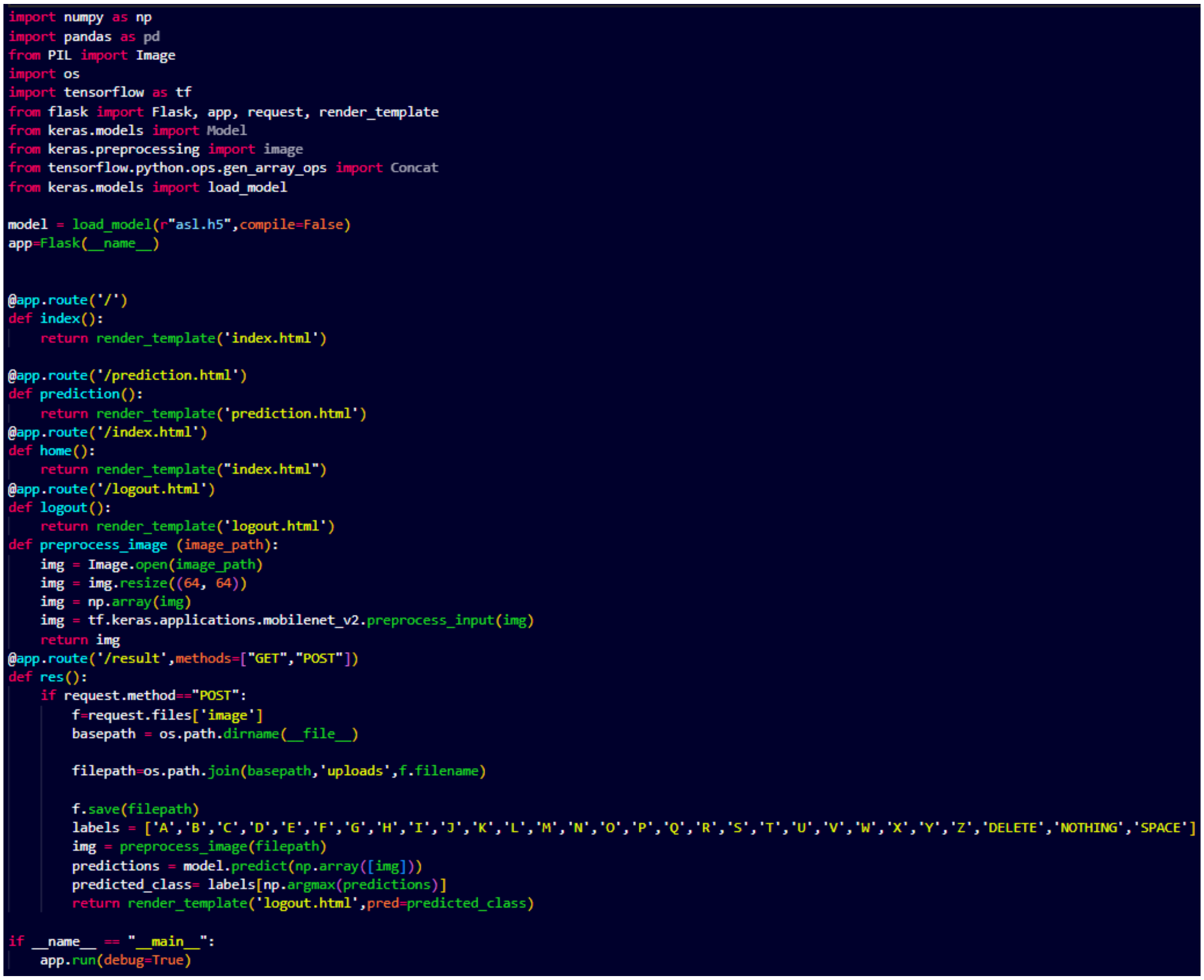


Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	5	2 Days	10 Nov 2023	12 Nov 2023	5	12 Nov 2023
Sprint-2	3	2 Days	12 Nov 2023	14 Nov 2023	3	14 Nov 2023
Sprint-3	3	2 Days	14 Nov 2023	16 Nov 2023	3	16 Nov 2023
Sprint-4	3	1 Days	16 Nov 2022	17 Nov 2023	3	17 Nov 2023

7. CODING & SOLUTIONING

7.1 Feature



```
import numpy as np
import pandas as pd
from PIL import Image
import os
import tensorflow as tf
from flask import Flask, app, request, render_template
from keras.models import Model
from keras.preprocessing import image
from tensorflow.python.ops.gen_array_ops import Concat
from keras.models import load_model

model = load_model(r"asl.h5",compile=False)
app=Flask(__name__)


@app.route('/')
def index():
    return render_template('index.html')

@app.route('/prediction.html')
def prediction():
    return render_template('prediction.html')
@app.route('/index.html')
def home():
    return render_template("index.html")
@app.route('/logout.html')
def logout():
    return render_template('logout.html')
def preprocess_image (image_path):
    img = Image.open(image_path)
    img = img.resize((64, 64))
    img = np.array(img)
    img = tf.keras.applications.mobilenet_v2.preprocess_input(img)
    return img
@app.route('/result',methods=["GET","POST"])
def res():
    if request.method=="POST":
        f=request.files['image']
        basepath = os.path.dirname(__file__)

        filepath=os.path.join(basepath,'uploads',f.filename)

        f.save(filepath)
        labels = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z','DELETE','NOTHING','SPACE']
        img = preprocess_image(filepath)
        predictions = model.predict(np.array([img]))
        predicted_class= labels[np.argmax(predictions)]
        return render_template('logout.html',pred=predicted_class)

if __name__ == "__main__":
    app.run(debug=True)
```

1. Model Loading and Initialization:

   This segment loads a pre-trained machine learning model (asl.h5) for ASL alphabet image recognition. The compile=False argument ensures that the model is loaded without recompilation.

2. Flask App Initialization:

   Initializes a Flask web application, defining the main application instance.

3. Routes for Web Pages:

Defines routes for different HTML pages (index.html, prediction.html, logout.html). These routes handle requests and render the respective HTML templates.

3. Image Preprocessing Function:

   Defines a function (preprocess_image) for preparing the input image before feeding it to the machine learning model. Resizing and preprocessing are essential for model compatibility.

4. Result Prediction Route:

   Handles image uploads, preprocesses the image, and uses the loaded model to predict the ASL alphabet sign. The result is then passed to the logout.html template for display.

5. Application Execution:

   Ensures that the Flask app runs when the script is executed directly, and the debug=True argument enables debugging mode for development.

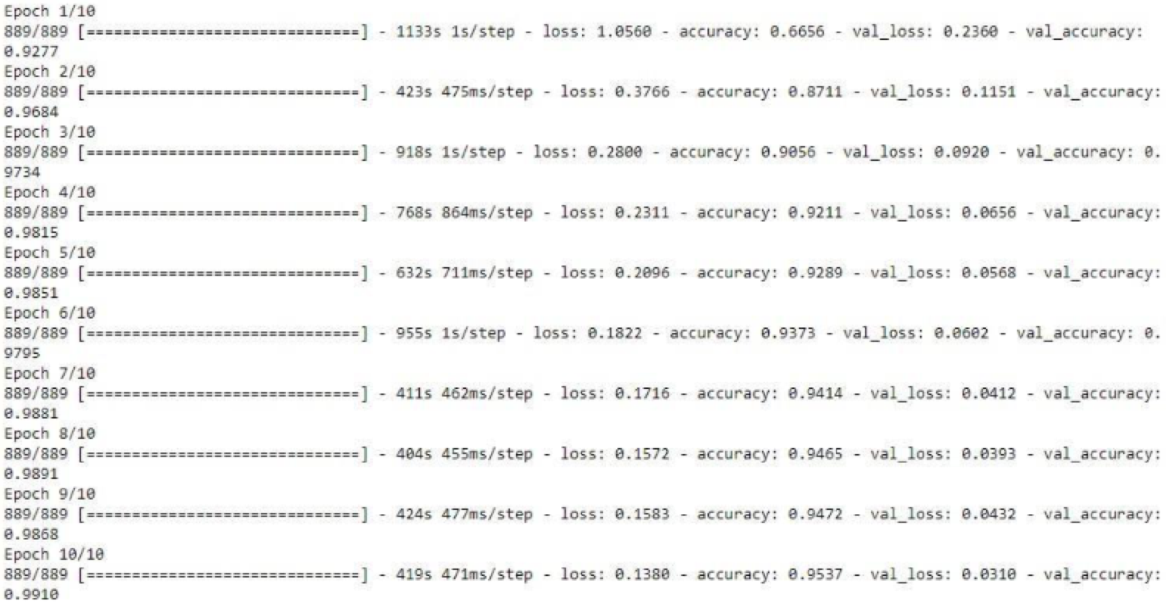
8. PERFORMANCE TESTING

8.1 Performance Metrics

Model Performance Testing:

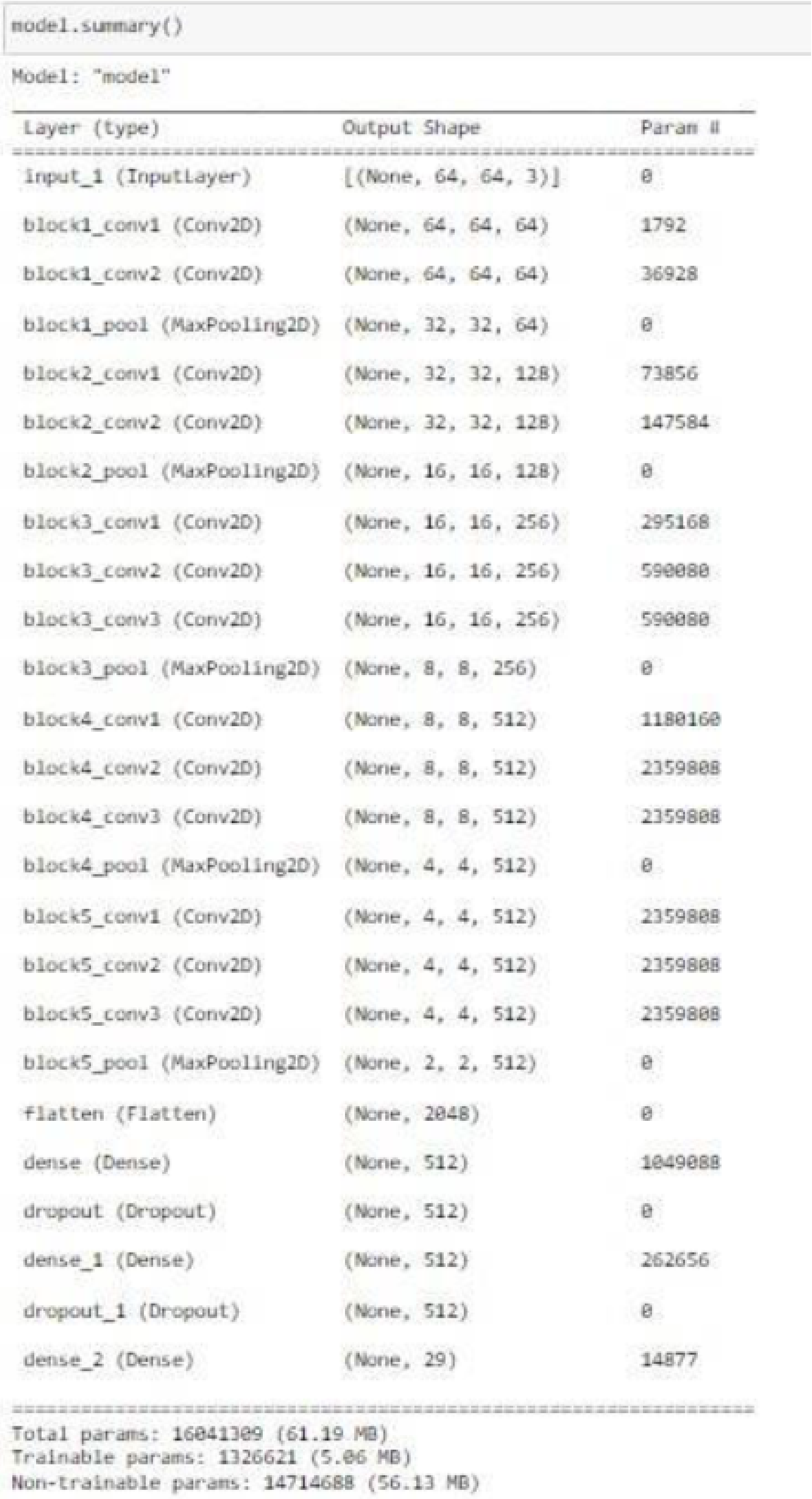
S.No.	Parameter	Values
1.	Model Summary	-
2.	Accuracy	Training Accuracy - 0.9537 Validation Accuracy - 0.9910

Training and Validation Accuracy Screenshot



```
Epoch 1/10
889/889 [==============================] - 1133s 1s/step - loss: 1.0560 - accuracy: 0.6656 - val_loss: 0.2360 - val_accuracy:
0.9277
Epoch 2/10
889/889 [==============================] - 423s 475ms/step - loss: 0.3766 - accuracy: 0.8711 - val_loss: 0.1151 - val_accuracy:
0.9684
Epoch 3/10
889/889 [==============================] - 918s 1s/step - loss: 0.2800 - accuracy: 0.9056 - val_loss: 0.0920 - val_accuracy: 0.
9734
Epoch 4/10
889/889 [==============================] - 768s 864ms/step - loss: 0.2311 - accuracy: 0.9211 - val_loss: 0.0656 - val_accuracy:
0.9815
Epoch 5/10
889/889 [==============================] - 632s 711ms/step - loss: 0.2096 - accuracy: 0.9289 - val_loss: 0.0568 - val_accuracy:
0.9851
Epoch 6/10
889/889 [==============================] - 955s 1s/step - loss: 0.1822 - accuracy: 0.9373 - val_loss: 0.0602 - val_accuracy: 0.
9795
Epoch 7/10
889/889 [==============================] - 411s 462ms/step - loss: 0.1716 - accuracy: 0.9414 - val_loss: 0.0412 - val_accuracy:
0.9881
Epoch 8/10
889/889 [==============================] - 404s 455ms/step - loss: 0.1572 - accuracy: 0.9465 - val_loss: 0.0393 - val_accuracy:
0.9891
Epoch 9/10
889/889 [==============================] - 424s 477ms/step - loss: 0.1583 - accuracy: 0.9472 - val_loss: 0.0432 - val_accuracy:
0.9868
Epoch 10/10
889/889 [==============================] - 419s 471ms/step - loss: 0.1380 - accuracy: 0.9537 - val_loss: 0.0310 - val_accuracy:
0.9910
```

Model Summary Screenshot



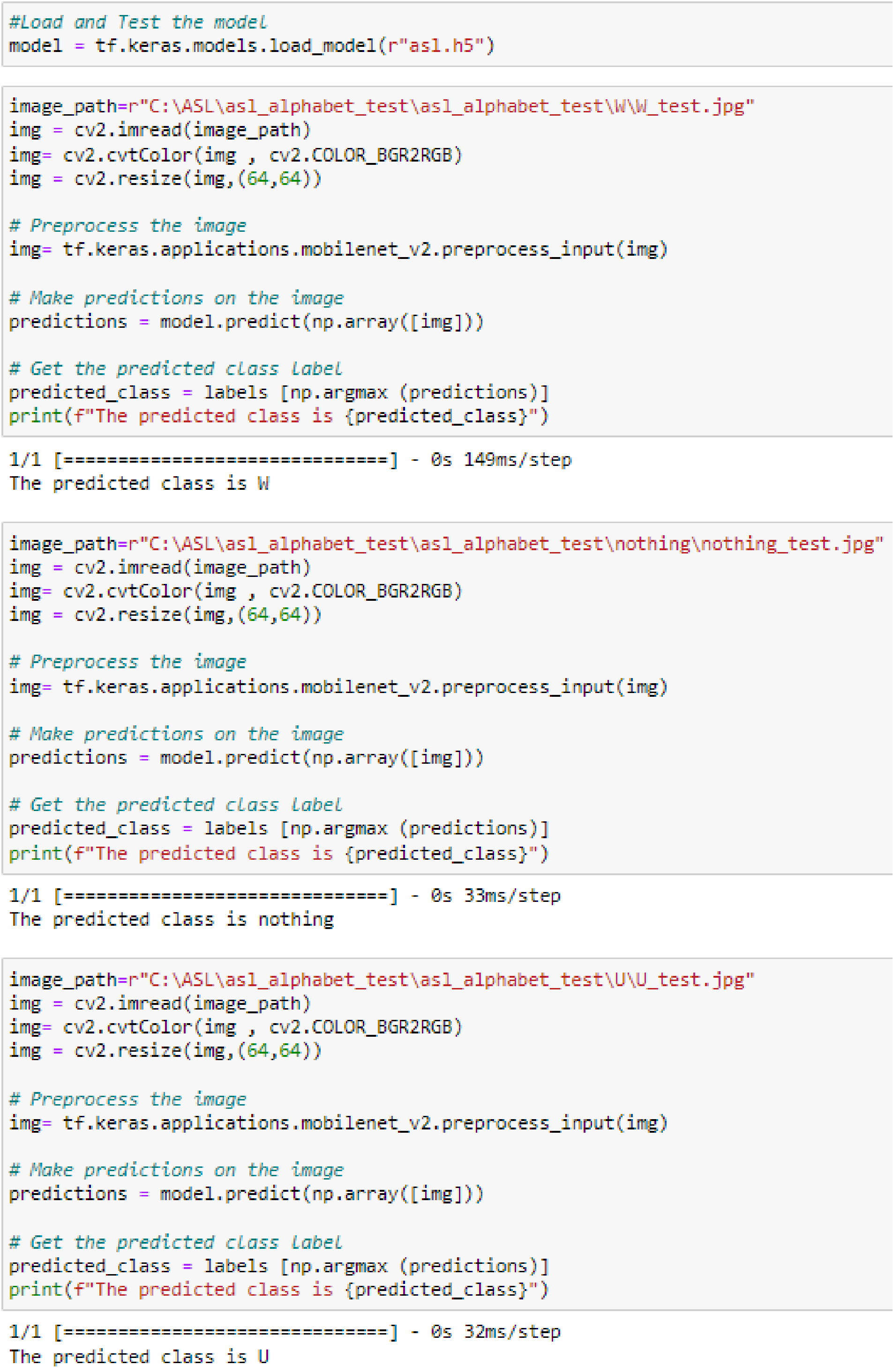
```
model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 64, 64, 3)]       0

 block1_conv1 (Conv2D)       (None, 64, 64, 64)        1792

 block1_conv2 (Conv2D)       (None, 64, 64, 64)        36928

 block1_pool (MaxPooling2D)  (None, 32, 32, 64)        0

 block2_conv1 (Conv2D)       (None, 32, 32, 128)       73856

 block2_conv2 (Conv2D)       (None, 32, 32, 128)       147584

 block2_pool (MaxPooling2D)  (None, 16, 16, 128)       0

 block3_conv1 (Conv2D)       (None, 16, 16, 256)       295168

 block3_conv2 (Conv2D)       (None, 16, 16, 256)       590080

 block3_conv3 (Conv2D)       (None, 16, 16, 256)       590080

 block3_pool (MaxPooling2D)  (None, 8, 8, 256)         0

 block4_conv1 (Conv2D)       (None, 8, 8, 512)         1180160

 block4_conv2 (Conv2D)       (None, 8, 8, 512)         2359808

 block4_conv3 (Conv2D)       (None, 8, 8, 512)         2359808

 block4_pool (MaxPooling2D)  (None, 4, 4, 512)         0

 block5_conv1 (Conv2D)       (None, 4, 4, 512)         2359808

 block5_conv2 (Conv2D)       (None, 4, 4, 512)         2359808

 block5_conv3 (Conv2D)       (None, 4, 4, 512)         2359808

 block5_pool (MaxPooling2D)  (None, 2, 2, 512)         0

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 512)               1049088

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 512)               262656

 dropout_1 (Dropout)         (None, 512)               0

 dense_2 (Dense)             (None, 29)                14877

=================================================================
Total params: 16041309 (61.19 MB)
Trainable params: 1326621 (5.06 MB)
Non-trainable params: 14714688 (56.13 MB)
```

9. RESULTS

9.1 Output Screenshots

Jupyter Notebook Prediction screenshots



```
#Load and Test the model
model = tf.keras.models.load_model(r"asl.h5")
```

```
image_path=r"C:\ASL\asl_alphabet_test\asl_alphabet_test\W\W_test.jpg"
img = cv2.imread(image_path)
img= cv2.cvtColor(img , cv2.COLOR_BGR2RGB)
img = cv2.resize(img,(64,64))

# Preprocess the image
img= tf.keras.applications.mobilenet_v2.preprocess_input(img)

# Make predictions on the image
predictions = model.predict(np.array([img]))

# Get the predicted class label
predicted_class = labels [np.argmax (predictions)]
print(f"The predicted class is {predicted_class}")
```

```
1/1 [==============================] - 0s 149ms/step
The predicted class is W
```

```
image_path=r"C:\ASL\asl_alphabet_test\asl_alphabet_test\nothing\nothing_test.jpg"
img = cv2.imread(image_path)
img= cv2.cvtColor(img , cv2.COLOR_BGR2RGB)
img = cv2.resize(img,(64,64))

# Preprocess the image
img= tf.keras.applications.mobilenet_v2.preprocess_input(img)

# Make predictions on the image
predictions = model.predict(np.array([img]))

# Get the predicted class label
predicted_class = labels [np.argmax (predictions)]
print(f"The predicted class is {predicted_class}")
```

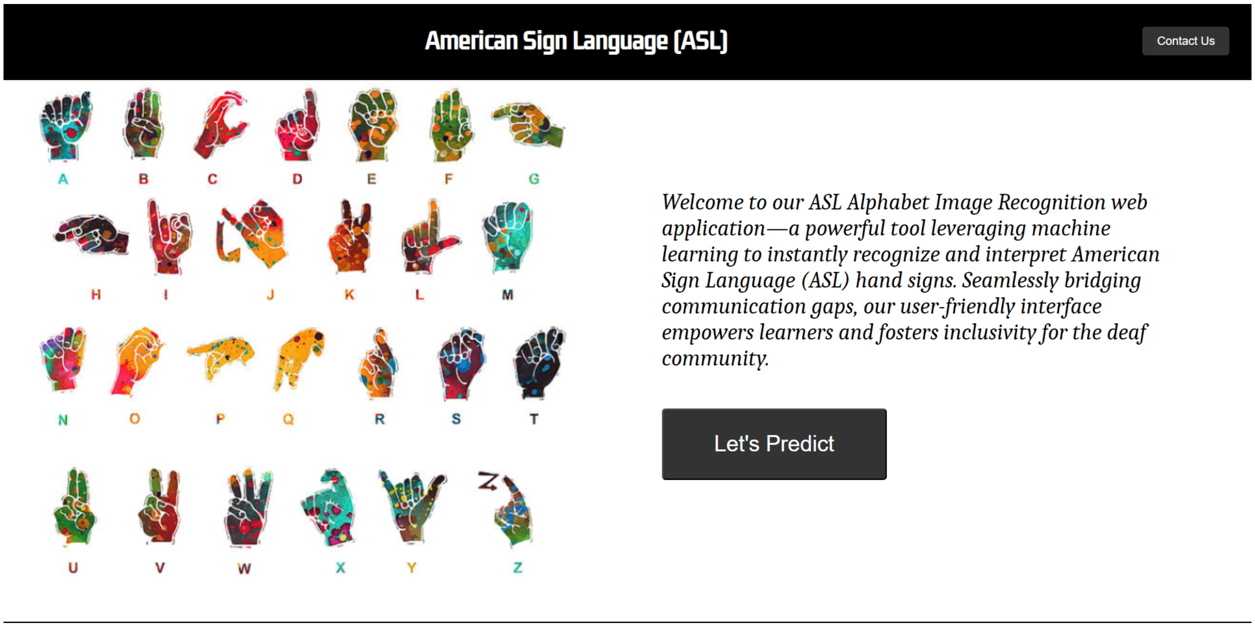
```
1/1 [==============================] - 0s 33ms/step
The predicted class is nothing
```

```
image_path=r"C:\ASL\asl_alphabet_test\asl_alphabet_test\U\U_test.jpg"
img = cv2.imread(image_path)
img= cv2.cvtColor(img , cv2.COLOR_BGR2RGB)
img = cv2.resize(img,(64,64))

# Preprocess the image
img= tf.keras.applications.mobilenet_v2.preprocess_input(img)

# Make predictions on the image
predictions = model.predict(np.array([img]))

# Get the predicted class label
predicted_class = labels [np.argmax (predictions)]
print(f"The predicted class is {predicted_class}")
```

```
1/1 [==============================] - 0s 32ms/step
The predicted class is U
```

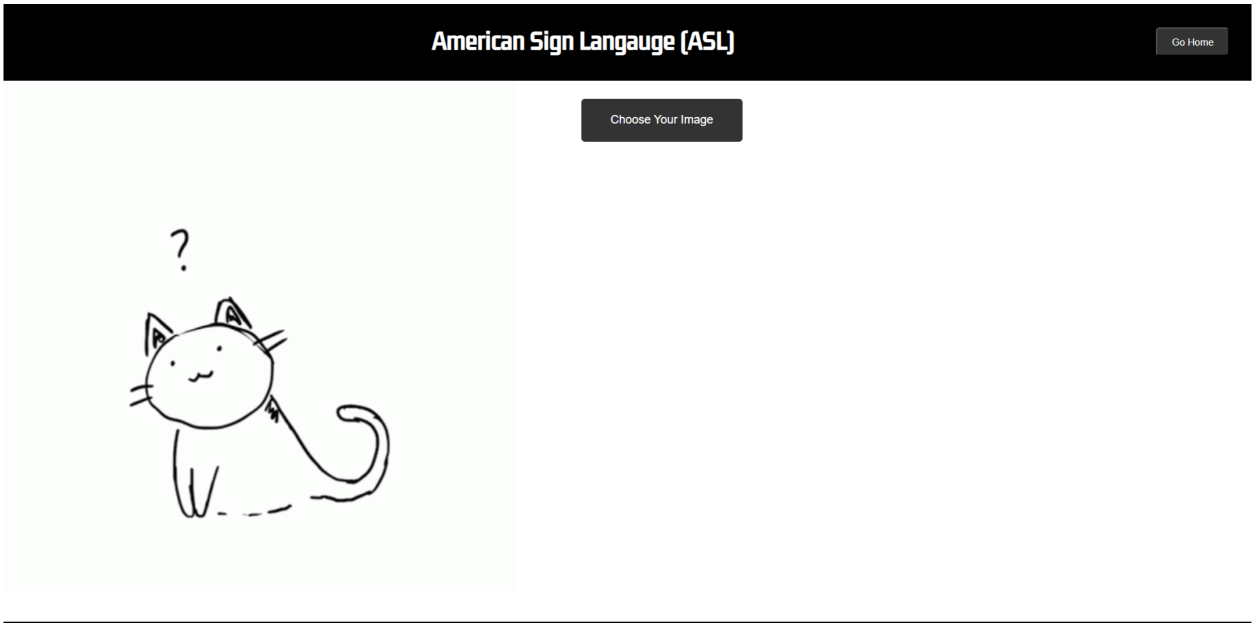
WebApp Screenshots

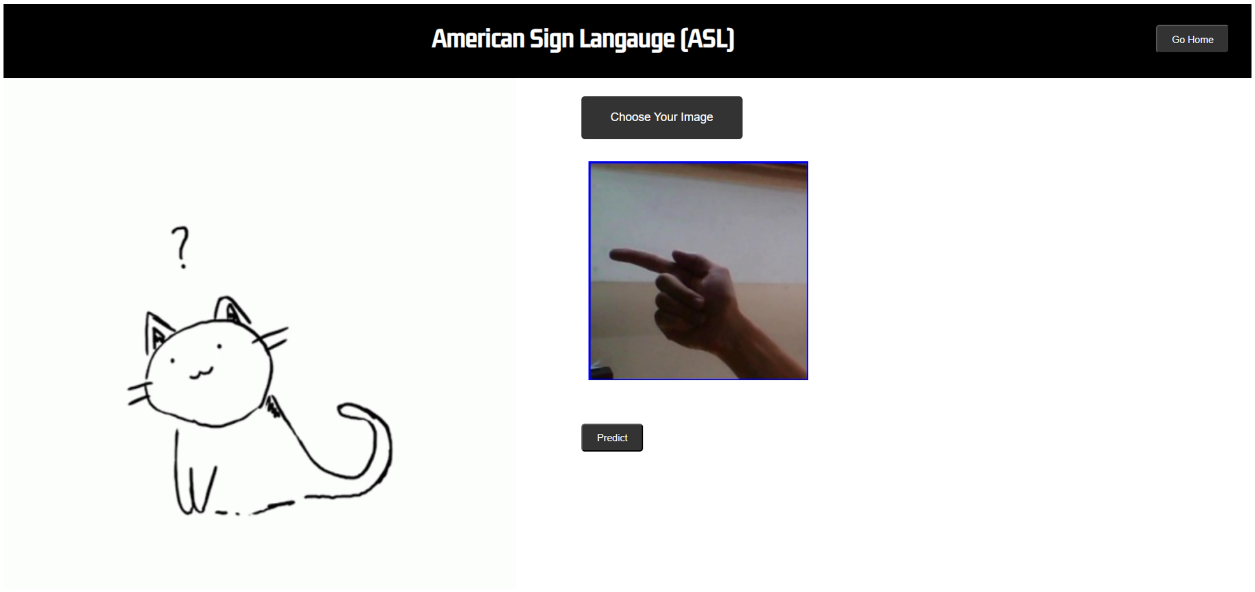
Index.html page



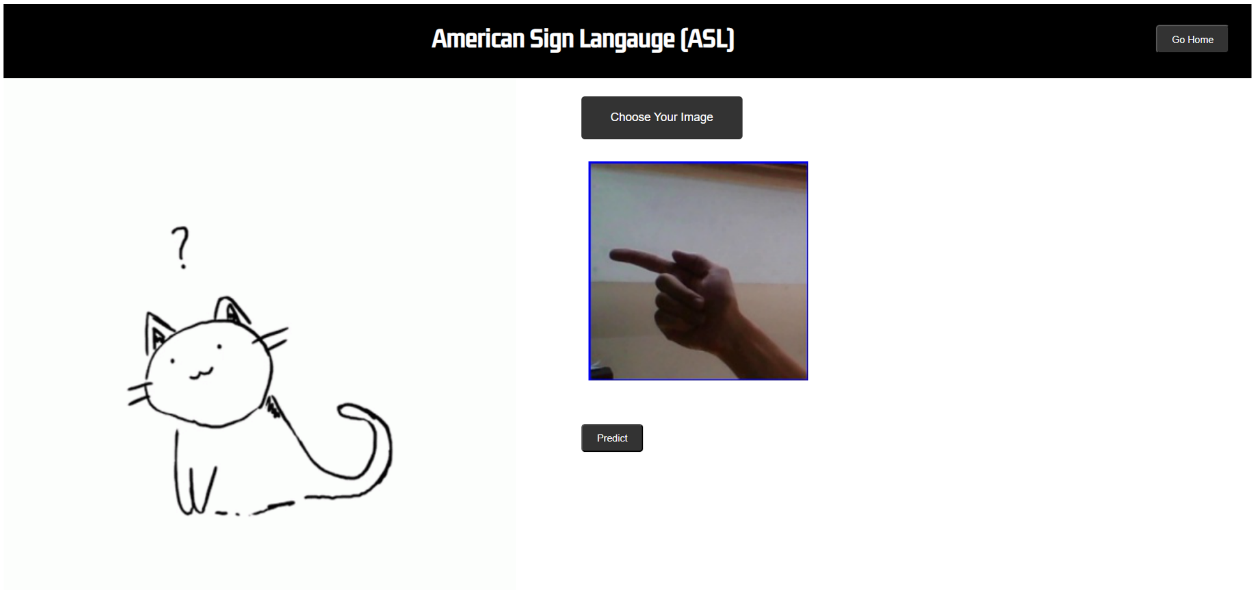
Welcome to our ASL Alphabet Image Recognition web application—a powerful tool leveraging machine learning to instantly recognize and interpret American Sign Language (ASL) hand signs. Seamlessly bridging communication gaps, our user-friendly interface empowers learners and fosters inclusivity for the deaf community.

prediction.html page





Logout.html (Result Page)



10.ADVANTAGES & DISADVANTAGES

Advantages:

- Communication Bridge:

  Facilitates communication between deaf and hearing communities.
- Real-Time Integration:

  Enables instant translation of ASL hand signs into text or spoken language.
- Educational Support:

  Assists in learning ASL by recognizing and explaining signs.
- Accessibility Boost:

  Enhances accessibility in various settings, promoting inclusivity.
- Technological Innovation:

  Demonstrates the application of AI-ML in assistive technology.

Disadvantages:

- Limited Dataset Variability:

  Generalization challenges with a less diverse training dataset.
- Ambiguity in Signs:

  Similar signs may lead to misclassifications.
- Resource Intensiveness:

  Computationally intensive for training and deployment.
- Limited Gesture Vocabulary:

  Designed for ASL alphabet and basic signs.

11.CONCLUSION

In summary, the ASL Alphabet Image Recognition project aims to bridge the communication gap for deaf individuals using American Sign Language. Leveraging machine learning and image recognition, the project focuses on real-time interpretation of ASL hand signs, including special signs for "space," "delete," and "nothing." This initiative not only addresses immediate communication challenges but also aspires to empower the deaf community, offering equal opportunities and fostering inclusivity. The project serves as a technological beacon towards a more accessible and connected society, with a commitment to breaking down barriers and making communication a universal right. Future refinements can further enhance scalability, security, and adaptability to evolving needs.

12.FUTURE SCOPE

The future scope of the ASL Alphabet Image Recognition project includes expanding gesture recognition beyond the alphabet, incorporating multimodal features, and enabling real-time translation. The development of mobile applications, user personalization, and integration with assistive devices is envisioned. The project could extend into educational tools, offer global language support for various sign languages, and focus on continuous model improvement through user feedback. Community engagement and collaboration with deaf communities are vital aspects for ensuring the system's relevance and effectiveness. These future directions aim to enhance the project's impact on communication accessibility and inclusivity for deaf individuals.

GitHub & Project Demo Link

Github - Link

Youtube Demo - Link