DOCUMENTATION

Real-Time Communication System Powered by AI for Specially Abled

TEAM ID: PNT2022TMID06610

TEAM MEMBERS:

1. BALAPATHY. V
2. SURYA NARAYANAN. S
3. GOUTHAM RAJ. K
4. SANTHOSH. S

Table of Contents

1. INTRODUCTION:

1.1 Project Overview:

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language. The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

1.2 Purpose:

As per our Indian Constitution, Article 51A (h),

"It shall be the duty of every citizen of India to develop the scientific temper, humanism and the spirit of inquiry and reform"

Accordingly, we, the future engineers of this great nation, have developed this project with the following purposes:

- Bridge the gap between specially-abled people and the normal people.
- To use technology to address the problems faced by the society.
- To contribute to the inclusive growth of the society.
- To make a positive impact/change in the society we live in.

2. LITERATURE SURVEY:

2.1 Existing Problems:

1. AAWAAZ: A Communication System for Deaf & Dumb by Anchal Sood, Anju Mishra (2016) [1]

Abstract: The paper proposes a framework for recognizing hand gesture which would serve not only as a way of communication between deaf and dumb and mute people, but also, as an instructor. Deaf and dumb individuals lack in proper communication with normal people and find it difficult to properly express themselves. Thus, they are subjected to face many issues in this regard. The sign language is very popular among them and they use it to express themselves. Thus, there is a need of a proper translator. The deaf and dumb are not idle as past, they are working outside and doing great at it. So, an efficient system must be set up, to interact with them, to know their views and ideas. The framework here, act as a communication system for deaf and dumb individuals. It would take the sign language as an input which would display the result not only in the form of text but also in the form of audio. Similarly, if there is any input in the form of text, it would display the corresponding image.

Methodology: From the input RGB image, the hand is separated and morphological operations are performed to identify the region of interest. The features of the gesture are then extracted and compared to a database of features of standard gestures. Finally, based on the comparison the output is generated.

Limitations: The proposed framework is good for recognizing hand gestures. But it is not feasible in every environment.

2. Full Duplex Communication System for Deaf & Dumb People by Shraddha R. Ghorpade, Surendra K. Waghamare (2015) [2]

Abstract: One of the important problems that our society faces is that people with disabilities are finding it hard to cope-up with the fast-growing technology. The access to communication technologies has become essential for the handicapped people. Generally deaf and dumb people use sign language for communication but they find difficulty in communicating with others who don't understand sign language. Sign language is an expressive and natural way for communication between normal and dumb people (information majorly conveyed through the hand gesture). So, we need a translator to understand what

they speak and communicate with us. The sign language translation system translates the normal sign language to speech and hence makes the communication between normal person and dumb people easier. But the question arises, how the deaf person understands the speech of a normal person and hence we need a system which converts the speech of normal person to text and the corresponding gesture is displayed on display. So, the whole idea is to build a device that enables two-way communications between deaf-mute person and a normal person.

Methodology: The methodology used is similar to [1] except that, instead of bare hands, the system requires the user to wear gloves to extract hand gesture.

Limitations: Gloves are mandatory. Without them, the system would not work. It is not feasible to carry gloves all the time. These are expensive as well.

3. Sign Language Recognition System by Er. Aditi Kalsh, Dr N.S. Garewal (2013) [3]

Abstract: Communication is the process of exchanging information, views and expressions between two or more persons, in both verbal and non-verbal manner. Hand gestures are the non-verbal method of communication used along with verbal communication. A more organized form of hand gesture communication is known as sign language. In this language each alphabet of the English vocabulary is assigned a sign. The physically disabled person like the deaf and the dumb uses this language to communicate with each other. The idea of this project is to design a system that can understand the sign language accurately so that the less fortunate people may communicate with the outside world without the need of an interpreter. By keeping in mind, the fact that in normal cases every human being has the same hand shape with four fingers and one thumb, this project aims at designing a real time system for the recognition of some meaningful shapes made using hands.

Methodology: The image is converted into grayscale and the edges of the fingers are detected using Canny edge detection. Then using the detected finger tips the gesture is recognized.

Limitations: The background of an image must be free from external objects. Also, the distance between the image and the camera is kept fixed.

4. Intelligent Sign Language Recognition Using Image Processing by Sawant Pramada, Deshpande Saylee, NalePranita, NerkarSamiksha, Mrs. Archana S. Vaidya (2013) [4]

Abstract: Computer recognition of sign language is an important research problem for enabling communication with hearing impaired people. This project introduces an efficient and fast algorithm for identification of the number of fingers opened in a gesture representing an alphabet of the Binary Sign Language. The system does not require the hand to be perfectly aligned to the camera. The project uses image processing system to identify, especially English alphabetic sign language used by the deaf people to communicate. The basic objective of this project is to develop a computer based intelligent system that will enable dumb people significantly to communicate with all other people using their natural hand gestures. The idea consisted of designing and building up an intelligent system using image processing, machine learning and artificial intelligence concepts to take visual inputs of sign language's hand gestures and generate easily recognizable form of outputs. Hence the objective of this project is to develop an intelligent system which can act as a translator between the sign language and the spoken language dynamically and can make the communication between people with hearing impairment and normal people both effective and efficient. The system is we are implementing for Binary sign language but it can detect any sign language with prior image processing.

Methodology: The RGB image is converted into a binary image. Certain coordinates are mapped to the binary image. Using a pattern matching algorithm the coordinates are then compared to the coordinates in a database. Based on the comparison, the gesture is identified.

Limitations: There is a specific camera orientation and specification. Thus, it is not feasible in every environment.

2.2 References:

[1] Sood Anchal, and Anju Mishra, "AAWAAZ: A communication system for deaf and dumb," 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). IEEE, 2016.

[2] Shraddha R. Ghorpade, Surendra K. Waghamare, "Full Duplex Communication System for Deaf & Dumb People," International Journal of

Emerging Technology and Advanced Engineering (IJETAE), Volume 5, Issue 5, May 2015, ISSN 2250-2459.

[3] Er. Aditi Kalsh, Dr N.S. Garewal, "Sign Language Recognition System," International Journal of Computational Engineering Research (IJCER), Volume 03, Issue 6, June 2013.

[4] Sawant Pramada, Deshpande Saylee, NalePranita, NerkarSamiksha, Mrs.Archana S. Vaidya "Intelligent Sign Language Recognition Using Image Processing," IOSR Journal of Engineering (IOSRJEN), Volume 3, Issue 2, Feb. 2013, PP 45-51.

2.3 Problem Statement Definition:

Problem Statement (PS)	I am	I'm trying to	But	Because	Which makes me feel
PS-1	Singh (a deaf-mute person)	Communicate with other (normal people)	I find it difficult to express/ communicate	They don't understand my sign language	Left alone
PS-2	Sita (a deaf-mute person)	Express my feelings to others	I can't do so like the normal people	I can't communicate in a way they can easily comprehend	Depressed and sad
PS-3	John (a deaf-mute person)	Reach a place in a new city	I can't ask for directions/help from others	They can't understand my sign language well	Insecure and worried
PS-4	Ahmed (a deaf-mute person)	Complain about an issue to a public authority	I find it difficult to do so	The public authority finds it difficult to understand me	Inferior and incompeten t

3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas:

3.2 Ideation & Brainstorming:

Problem Statement

A deaf-mute person is trying to communicate with other normal people, but he/she finds it difficult to do so because the rest of them cannot understand their sign language with the exception of a few, which makes him/her feel left alone, depressed, sad, inferior, insecure and incompetent.

2

Brainstorm

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

⏱ 10 minutes

SURYA NARAYANAN

- It should use deep-learning to translate deaf-mute sign language
- Our product should help the deaf-mute to communicate with normal people
- Use neural networks to translate the sign language
- Input: Sign language Output: Translated normal language

GOUTHAM RAJ

- It should have a easy to use GUI
- Solution should be made universally accessible
- Single Application
- The cost should be less

SANTHOSH

- Translation error should be as low as possible
- Help the deaf-mute people mentally and emotionally
- Bridge the gap between deaf-mute and normal people
- It should convert sign language in a way that should be easy to grasp for normal people

BALAPATHY

- Should work with existing hardware
- Our product should be user-friendly
- The deaf-mute should be able to use our product confidently
- Improve the confidence of deaf-mute people

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

SOLUTION SPECIFIC

- It should use deep-learning to translate deaf-mute sign language
- Single Application
- Translation error should be as low as possible
- Use neural networks to translate the sign language
- Input: Sign language Output: Translated normal language
- It should have a easy to use GUI

USABILITY

- Our product should be user-friendly
- The deaf-mute should be able to use our product confidently
- It should convert sign language in a way that should be easy to grasp for normal people

IT SHOULD SOLVE

- Help the deaf-mute people mentally and emotionally
- Bridge the gap between deaf-mute and normal people
- Our product should help the deaf-mute to communicate with normal people
- Improve the confidence of deaf-mute people

CONSIDERATIONS

- Should work with existing hardware
- The cost should be less
- Solution should be made universally accessible

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.

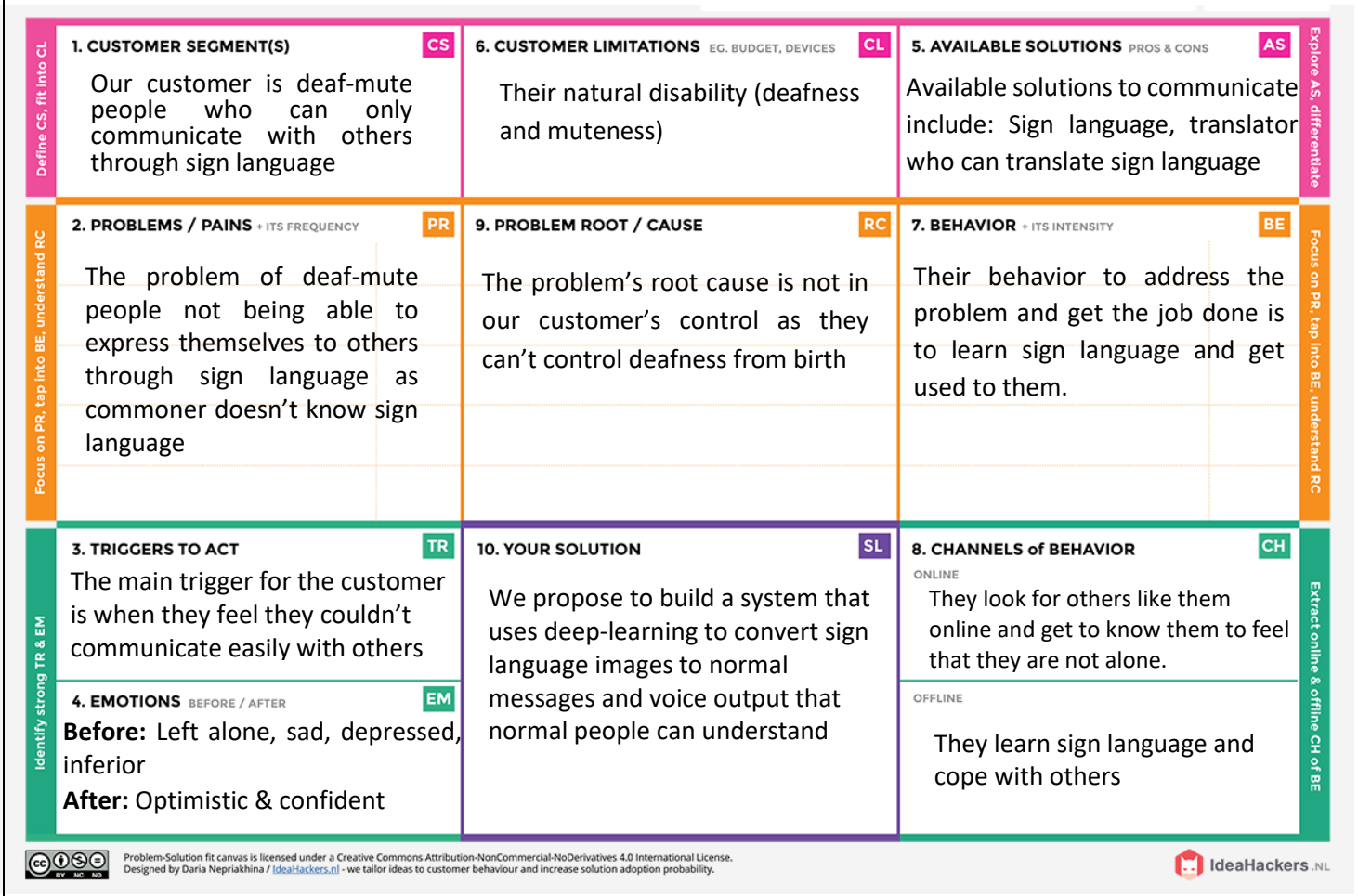
20 minutes

3.3 Proposed Solution:

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A deaf-mute person is trying to communicate with other normal people, but he/she finds it difficult to do so because the rest of them cannot understand their sign language with the exception of a few, which makes him/her feel left alone, depressed, sad, inferior, insecure and incompetent.
2.	Idea / Solution description	To design a system that used deep-learning to convert deaf-mute's sign language input to normal message and voice, so that, normal people can easily understand the deaf-mute.
3.	Novelty / Uniqueness	This idea is unique as it includes artificial intelligence (deep learning, etc) to solve the problem of deaf-mute communication. Also, it can result in higher accuracy of translation.
4.	Social Impact / Customer Satisfaction	The social impact of the solution is huge. It can create a positive and optimistic society for the deaf-mute to live in. The deaf-mute people may be immensely satisfied and overjoyed when they could communicate with normal people.
5.	Business Model (Revenue Model)	The solution has a business potential as it can tap the deaf-mute people's market. Also, public offices can use

		this system for their communication with deaf-mute people.
6.	Scalability of the Solution	The solution can be scaled easily to include new features and functionalities, and to cover wider range of people in the future.

3.4 Problem Solution Fit:



4. REQUIREMENT ANALYSIS:

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement	Sub Requirement (Sub-Task)
FR-1	**User Input**	The system gets the sign language input. The input may be image, video or live feed (depending upon the scope of the project)
FR-2	**Processing**	The system based on the trained model, should output the corresponding normal message.
FR-3	**System Output**	The system should output to the users, the normal message and voice.

4.2 Non-Functional Requirements:

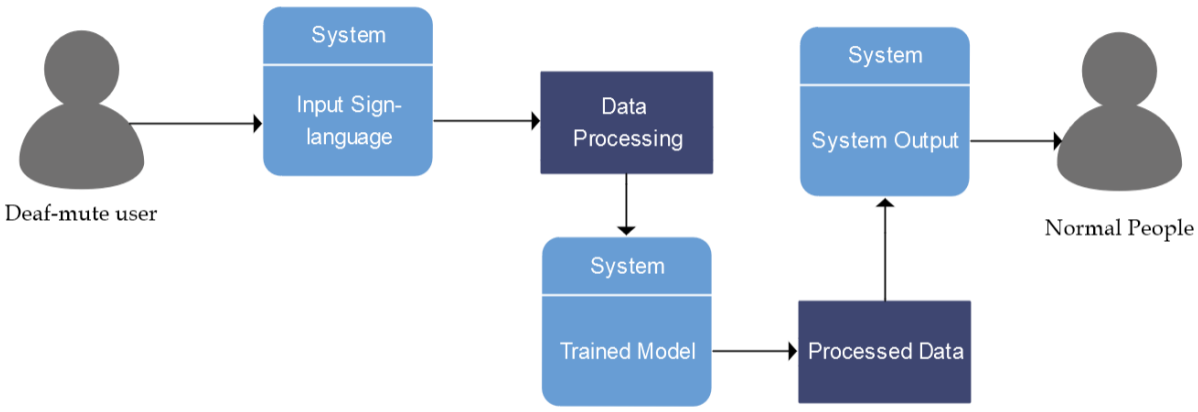
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Deaf-mute people should be able to use the system with ease. The same applies for normal people who get the system's output. The system should have good UI.
NFR-2	**Security**	Even though the use-case of the system doesn't need any security feature, it must be ensured that the privacy of user data be maintained and handled appropriately.

NFR-3	**Reliability**	The translation of sign languages should be reliable. The accuracy of the system should be tested extensively to make sure that it is up to the mark.
NFR-4	**Performance**	The processing should be done in a considerable time so the conversation can go on without waiting for the system's output.
NFR-5	**Availability**	The system should be universally accessible. Since sign language is almost same everywhere, the system can be used across the globe.
NFR-6	**Scalability**	The system should be scalable to accommodate new features and functionalities and to cater wider range of people in future.

5. PROJECT DESIGN:

5.1 Data Flow Diagrams:



5.2 Solution & Technical Architecture:

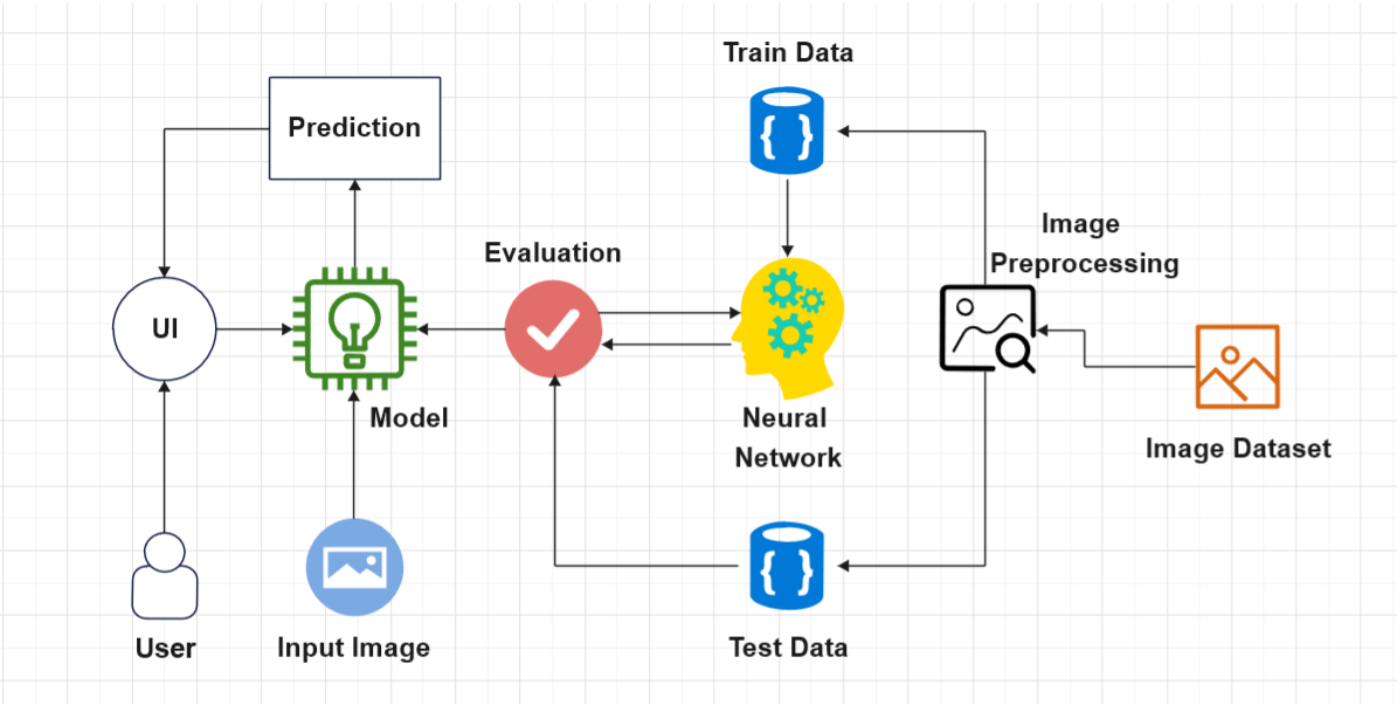


Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User (Deaf-mute)	The deaf-mute user will benefit from the system which uses several technologies.	Cloud tech, OpenCV and AI tech like Machine Learning, Deep Learning, etc.
2.	User Interface	The user interface lets the user interact with the system which is hosted in the cloud.	Suitable UI Technology, Cloud Hosting
3.	Models	A machine learning model is used to classify our gestureimage dataset.	Machine Learning
4.	Image Prediction	The image prediction is done with the help of deep learning which implements neural networks of various kinds to solve the problem.	ANN, CNN
5.	Image	Image processing is done on input image.	OpenCV
6.	Speech	The output of the system is speech (voice) to be heard fornormal users.	Suitable Speech System

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Our system implements many open-source frameworks.	AI frameworks, OpenCV, Speech System, UI system, Python Language
2.	Security Implementations	Necessary security measures will be implementedin the system.	Necessary Security Technologies

3.	Scalable Architecture	The architecture is very much scalable to accommodate any future needs.	Scalable Technologies
4.	Availability	The system will be made ubiquitous so that it is available everywhere.	Necessary Technologies
5.	Performance	The model will be fine-tuned to strike a balance between accuracy vs performance.	Optimization of code and trained model

5.3 User Stories:

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Customer (Deaf-Mute people)	User Input	USN - 1	As a user, I can input my sign-language to the system for processing.	The user can input sign-language into the system	Low	Sprint-3
		USN - 2	As a user, I can input sign-language images to the system for processing.	The user can input images into the system	High	Sprint-1
		USN - 3	As a user, I can make sure the input is captured correctly by the system.	The system should capture the input correctly	Medium	Sprint-2

	Processing	USN - 4	As a user, I can ensure that the sign-language input is correctly getting translated into normal message and voice.	The user can ensure that the processing is done correctly.	Medium	Sprint-2
		USN - 5	As a user, I can get acknowledgement from the system about the processing of the input.	The user should get an acknowledgement	High	Sprint-1
		USN - 6	As a user, I will get feedback about the processing of the system.	The user should get feedback from the system	Low	Sprint-3
	System Output	USN - 7	As a user, I can acknowledge the output of the system by ensuring messages are displayed.	The user should get an acknowledgement from the system	High	Sprint-1
		USN - 8	As a user, I can get feedback about the system from its output.	The user should get feedback from the system	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation:

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	User Input	USN-1	As a user, I can input my sign-language to the system for processing.	10	Low	Surya Narayanan, Goutham Raj
Sprint-1	User Input	USN-2	As a user, I can input sign-language images to the system for processing.	10	High	Balapathy, Surya Narayanan
Sprint-2	User Input	USN-3	As a user, I can make sure the input is captured correctly by the system.	5	Medium	Surya Narayanan, Santhosh
Sprint-2	Processing	USN-4	As a user, I can ensure that the sign-language input is correctly getting translated into normal message and voice.	10	Medium	Surya Narayanan, Santhosh, Goutham Raj
Sprint-1	Processing	USN-5	As a user, I can get acknowledgement from the system about the processing of the input.	5	High	Balapathy, Surya Narayanan
Sprint-3	Processing	USN-6	As a user, I will get feedback about the processing of the system.	10	Low	Surya Narayanan, Goutham Raj
Sprint-1	System Output	USN-7	As a user, I can acknowledge the output of the system	5	High	Surya Narayanan, Balapathy

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			by ensuring messages are displayed.			
Sprint-2	System Output	USN-8	As a user, I can get feedback about the system from its output.	5	Medium	Surya Narayanan, Santhosh

Project Tracker, Velocity & Burndown Chart:

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

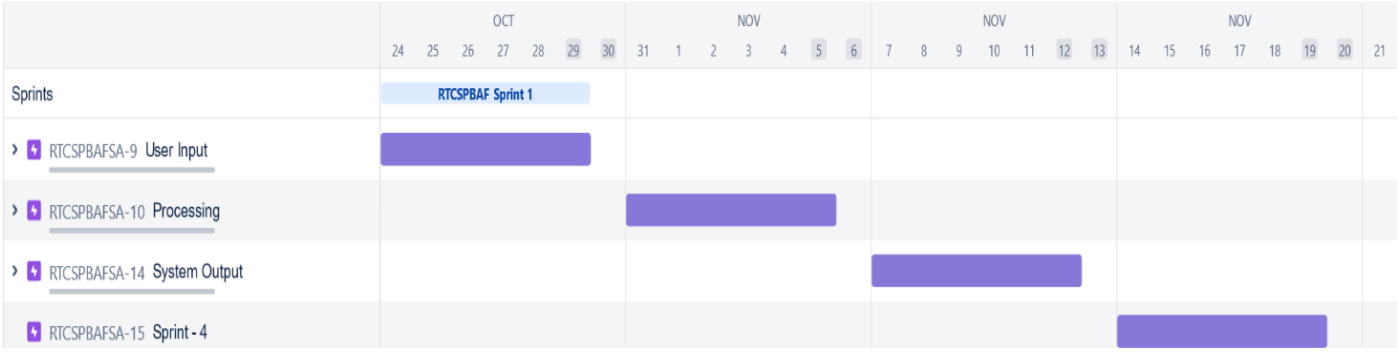
Velocity:

The team's average velocity (AV) per iteration unit (story points per day):

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

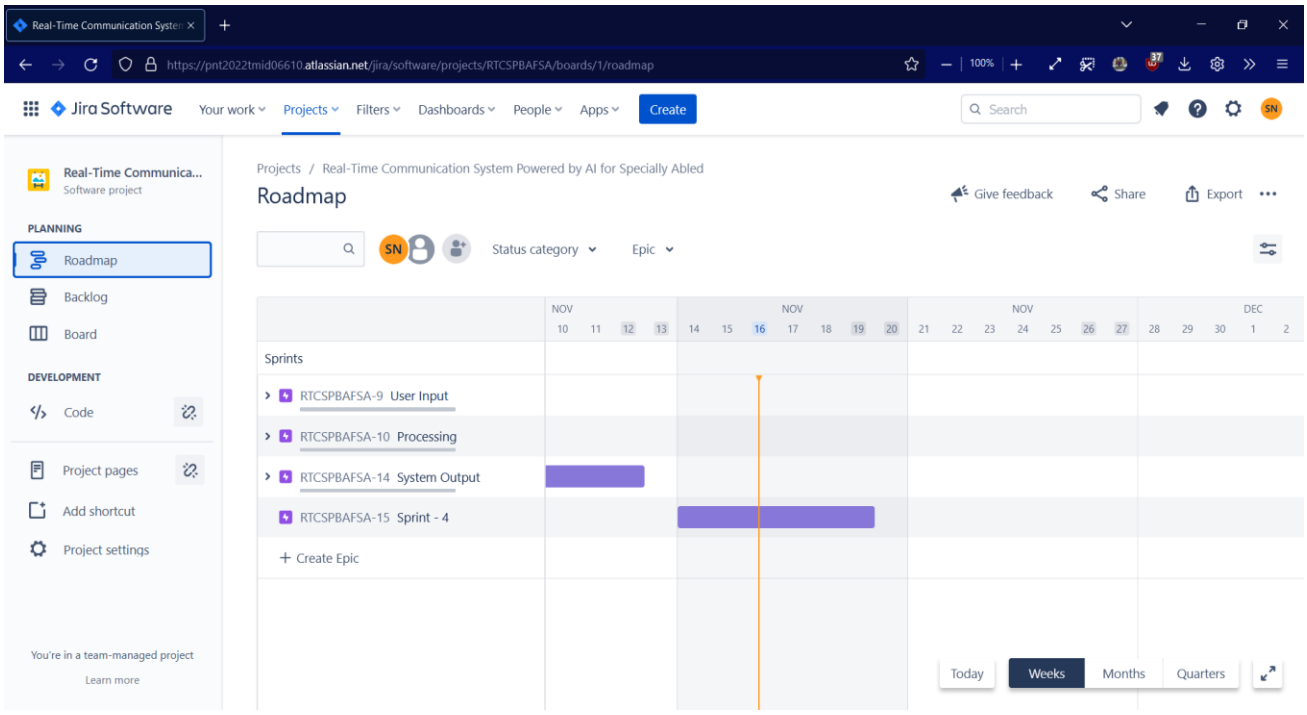


6.2 Sprint Delivery Schedule:

Title	Description	Date
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	28 September 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	24 September 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	25 September 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 September 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 September 2022
Solution Architecture	Prepare solution architecture document.	28 September 2022

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	20 October 2022
Functional Requirement	Prepare the functional requirement document.	8 October 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	9 October 2022
Technology Architecture	Prepare the technology architecture diagram.	10 October 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	22 October 2022
Project Development - Delivery of Sprint 1, 2, 3 & 4	Develop & submit the developed code by testing it.	19 November 2022

6.3 Reports From JIRA:



7. CODING & SOLUTIONING:

7.1 Feature 1 – Building the Model to Classify Sign Language:

In Feature – 1 of our project, we collect the dataset, pre-process it, create our CNN model, train the model, and save the trained model.

We have used a dataset from Kaggle.com to classify the ISL (Indian Sign Language) alphabets (A-Z).

Data Collection

Note: We use our custom dataset downloaded from kaggle.

```
[1]  !unzip '/content/drive/MyDrive/Dataset.zip'
```
Python

We split the dataset into training dataset and testing dataset. Then, we build a Convolutional Neural Network (CNN) model to classify the training dataset. Finally, the trained model is tested using the testing dataset.

ImageDataGenerator for training and testing set

```
16]  train_datagen = ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True)
```
Python

```
17]  test_datagen = ImageDataGenerator(rescale=1./255)
```
Python

```
training_data = train_datagen.flow_from_directory('/content/Dataset/training_set',
                                                  target_size=(128,128),
                                                  class_mode='categorical',
                                                  batch_size=100, color_mode='grayscale')
```

Importing ImageDataGenerator

Note: We use our custom dataset downloaded from kaggle.

```
5]  from tensorflow.keras.preprocessing.image import ImageDataGenerator
    import matplotlib.pyplot as plt
    import cv2
```
Python

```python
training_data.class_indices #All categories in training data
```

```python
testing_data.class_indices # All categories in testing data
```

Model Building

Import the required model building libraries

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Flatten, Dense, MaxPooling2D
```

For the Convolutional Neural Network (CNN), we used:

1. Input layer
2. Conv2D layer
3. Conv2D layer
4. Dense layer (Hidden layer)
5. Dense layer (Hidden layer)
6. Output layer

Initialize the model, Add convolution layer, pooling layer, flatten layer, dense layer

```python
model = Sequential() # Sequential Model

model.add(Conv2D(128, (3,3), activation='relu', input_shape=(128,128,1))) # Convolution Layer
model.add(MaxPooling2D(pool_size=(2,2))) # Pooling Layer

model.add(Conv2D(128, (3,3), activation='relu')) # Convolution Layer
model.add(MaxPooling2D(pool_size=(2,2))) # Pooling Layer

model.add(Flatten()) # Flatten Layer
model.add(Dense(128, activation='relu')) # Hidden Layer
model.add(Dense(128, activation='relu')) # Hidden Layer
model.add(Dense(26, activation='softmax')) # Hidden Layer
```

Compiler the model

```python
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Fit the model

```
model.fit_generator(training_data,
                    steps_per_epoch=len(training_data),
                    epochs=3,
                    validation_data=testing_data,
                    validation_steps=len(testing_data))
```

[25] Python

The model is trained for 3 epochs and the accuracy of the model is above 95%.

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version. Please use
`Model.fit`, which supports generators.
  """

Epoch 1/3
294/294 [==============================] - 76s 257ms/step - loss: 0.1688 - accuracy: 0.9566 -
val_loss: 0.0190 - val_accuracy: 0.9927
Epoch 2/3
294/294 [==============================] - 75s 254ms/step - loss: 0.0059 - accuracy: 0.9984 -
val_loss: 0.0012 - val_accuracy: 0.9996
Epoch 3/3
294/294 [==============================] - 75s 254ms/step - loss: 0.0163 - accuracy: 0.9958 -
val_loss: 1.3604e-04 - val_accuracy: 1.0000

<keras.callbacks.History at 0x7f097e74eb50>
```

Finally, the trained model is saved for future use.

Save the model

```
model.save('trained_model.h5')
```

[26] Python

7.2 Feature 2 – Building the Flask Application:

In Feature – 2 of our project, we create a Flask based web application to present it to the user.

The Flask application consists of:

- app.py (main script)
- backend.py (contains the trained model)
- index.html (frontend of the application)
- translate.html (frontend of the application)

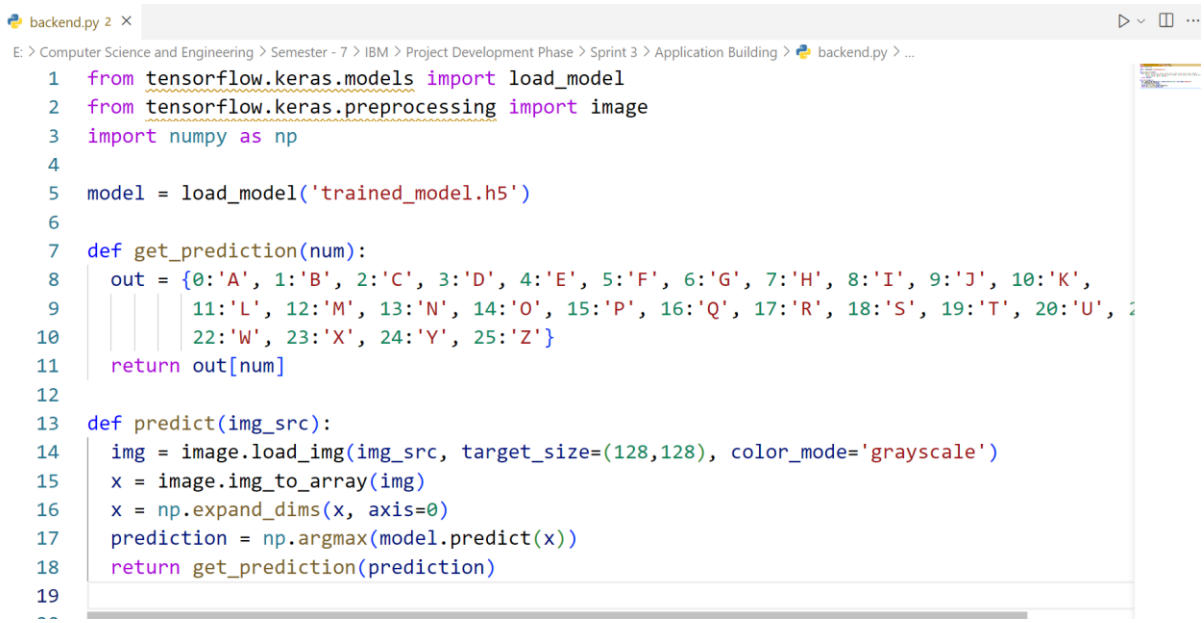
App.py:

This is the main script for the Flask application. This script communicates with backend.py and the frontend of the application.

```python
from flask import Flask, render_template, request, redirect, url_for
import backend

app = Flask(__name__)
path=''

@app.route('/', methods =["GET","POST"])
def index():
    global path
    if request.method == "POST":
        path = request.form.get("myInput")
        print(path)
        return render_template("translate.html", img_path=str(path), pred=str(backend.predi
    return render_template("index.html")


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

Backend.py

This is the script which contains the trained CNN model. It responds to the requests made from app.py.



```python
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

model = load_model('trained_model.h5')

def get_prediction(num):
  out = {0:'A', 1:'B', 2:'C', 3:'D', 4:'E', 5:'F', 6:'G', 7:'H', 8:'I', 9:'J', 10:'K',
         11:'L', 12:'M', 13:'N', 14:'O', 15:'P', 16:'Q', 17:'R', 18:'S', 19:'T', 20:'U',
         22:'W', 23:'X', 24:'Y', 25:'Z'}
  return out[num]

def predict(img_src):
  img = image.load_img(img_src, target_size=(128,128), color_mode='grayscale')
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)
  prediction = np.argmax(model.predict(x))
  return get_prediction(prediction)

```

Index.html:

This is the script for the frontend of the application. It contains the necessary code (HTML and CSS) to make the user interactable with the application.

index.html

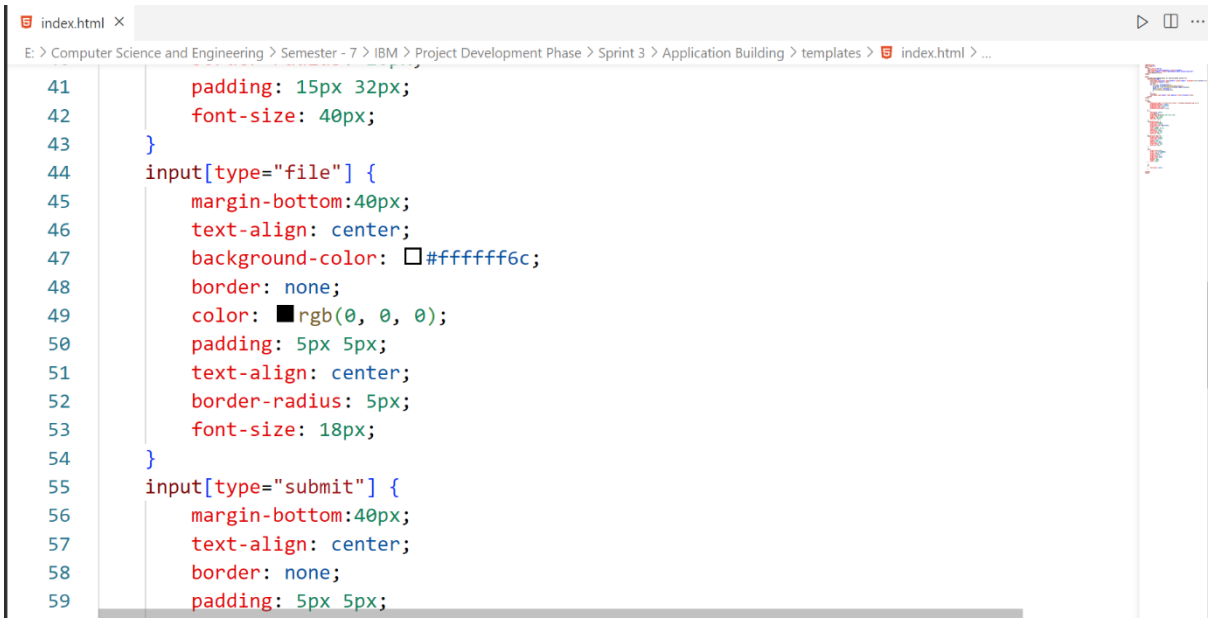
E: > Computer Science and Engineering > Semester - 7 > IBM > Project Development Phase > Sprint 3 > Application Building > templates > index.html > ...

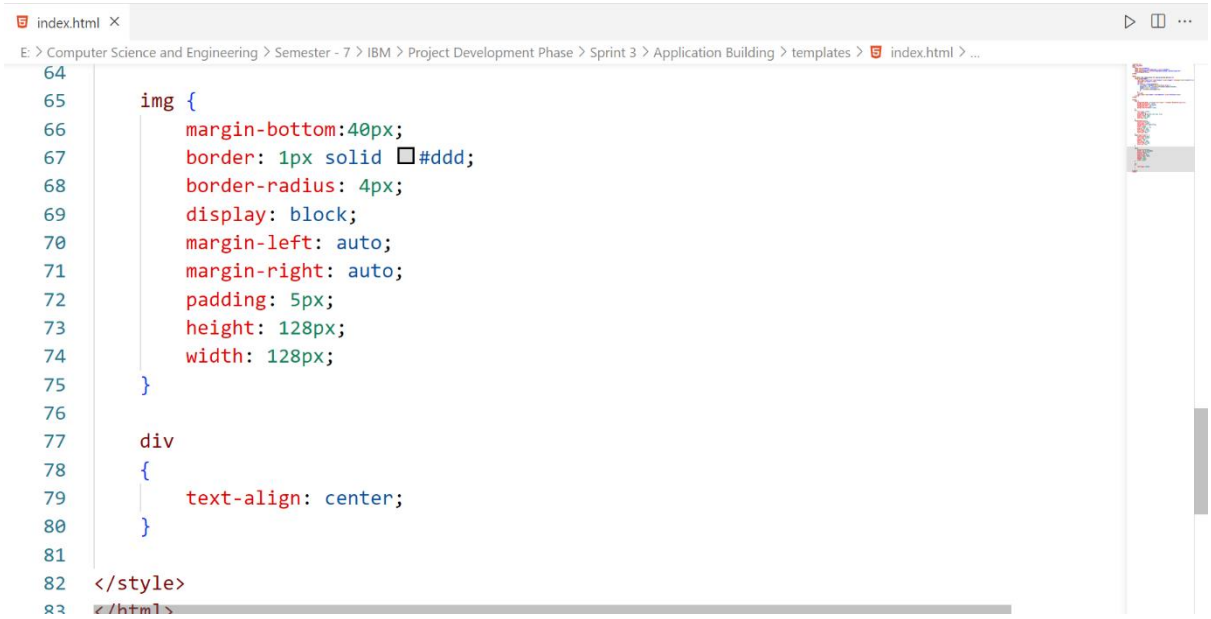
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Project</title>
</head>

<body>
    <h1>Real-time communication for specially-abled persons</h1>
    <form method="POST">
        <div><input type="file" name="myInput" accept="image/*" onchange="loadFile(event)"
        <div><img id="output"/></div>
        <script>
        var loadFile = function(event) {
            var output = document.getElementById('output');
            output.src = URL.createObjectURL(event.target.files[0]);
            output.onload = function() {
```

index.html

E: > Computer Science and Engineering > Semester - 7 > IBM > Project Development Phase > Sprint 3 > Application Building > templates > index.html > ...

```
            }
        };
        </script>
        <div><input type="submit" name="myButton" value="Translate"></div>
    </form>
</body>

<style>
    body {
        background-image: url("{{url_for('static', filename='Background.jpg')}}");
        background-position: center;
        background-repeat: no-repeat;
        background-size: cover;
        background-attachment: fixed;
    }
    h1 {
        text-align: center;
        color: black;
        background: rgba(255, 255, 255, 0.4);
```





Translate.html:

Another script similar to index.html is used which displays the translation. The changes in translate.html when compared to index.html is a single line of code:

```
    <div><strong>The ISL alphabet is: {{pred}}</strong></div>
</body>
```

8. TESTING:

8.1 Test Cases:

Testing of a built application is as important as building the application itself. Here, we test the built model with our testing dataset images.

Testing The Model

Making necessary imports

```
from tensorflow.keras.preprocessing import image
import numpy as np
from tensorflow.keras.models import load_model
import cv2
```

[8] Python

Load the Trained Model

```
model = load_model('/content/drive/MyDrive/trained_model.h5')
```

Python

Test Case – 1: ISL Alphabet ‘D’:

Predicting Test Images

Indian Sign Language(ISL) Alphabet - 'D'

```
img = image.load_img('/content/Dataset/test_set/D/1027.jpg', target_size=(128,128), color_mode=
img
```

6] Python

```
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
prediction = np.argmax(model.predict(x))
get_prediction(prediction)
```

[27] Python

```
1/1 [==============================] - 0s 57ms/step
'D'
```

Test Case – 2: ISL Alphabet ‘F’:

Indian Sign Language(ISL) Alphabet - 'F'

```
img = image.load_img('/content/Dataset/test_set/F/1022.jpg', target_size=(128,128), color_mode=
img
```

Python

+ Code + Markdown

```
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
prediction = np.argmax(model.predict(x))
get_prediction(prediction)
```

Python

```
1/1 [==============================] - 0s 49ms/step
'F'
```

+ Code + Markdown

Test Case – 3: ISL Alphabet ‘S’:

Indian Sign Language(ISL) Alphabet - 'S'

```
img = image.load_img('/content/Dataset/test_set/S/108.jpg', target_size=(128,128), color_mode='
img
```

Python

```
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
prediction = np.argmax(model.predict(x))
get_prediction(prediction)
```

Python

```
1/1 [==============================] - 0s 48ms/step
'S'
```

Test Case – 4: ISL Alphabet ‘C’:

We used our own image to simulate real-world working of the trained model.

Testing our own images in the model to simulate real-world usage

Our own test image - Indian Sign Language(ISL) Alphabet - 'C'

```
img = image.load_img('/content/drive/MyDrive/C.jpg', target_size=(128,128), color_mode='graysca
img
```

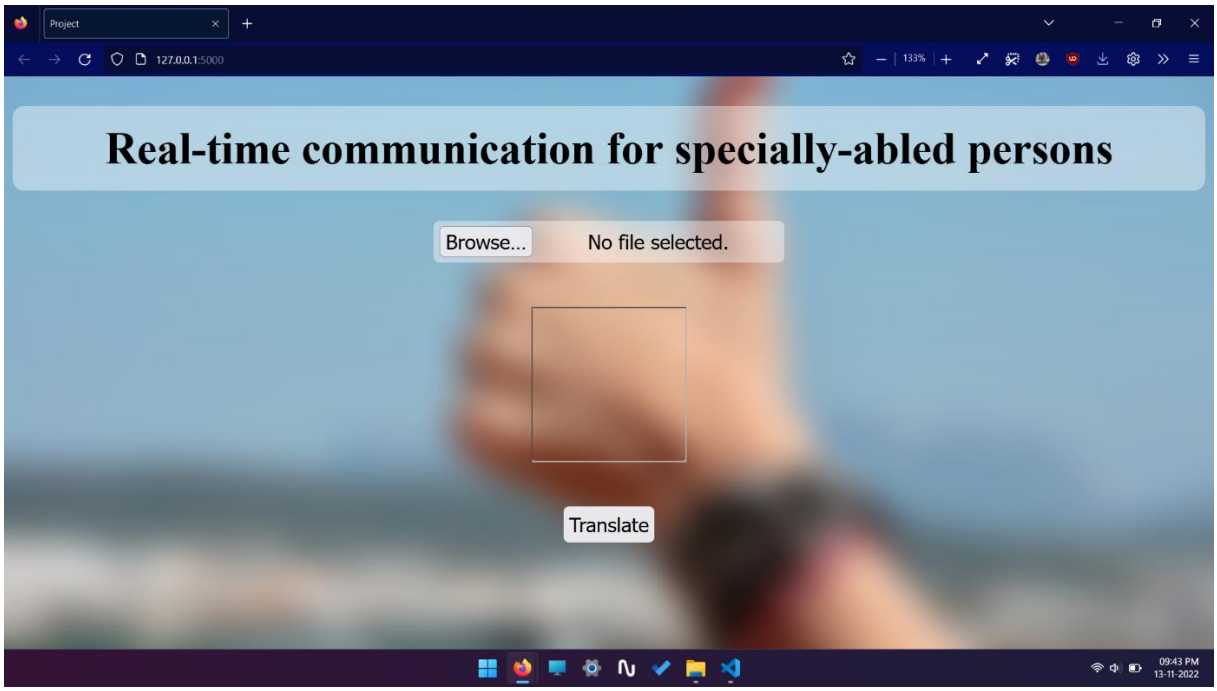
Python

```
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
prediction = np.argmax(model.predict(x))
get_prediction(prediction)
```

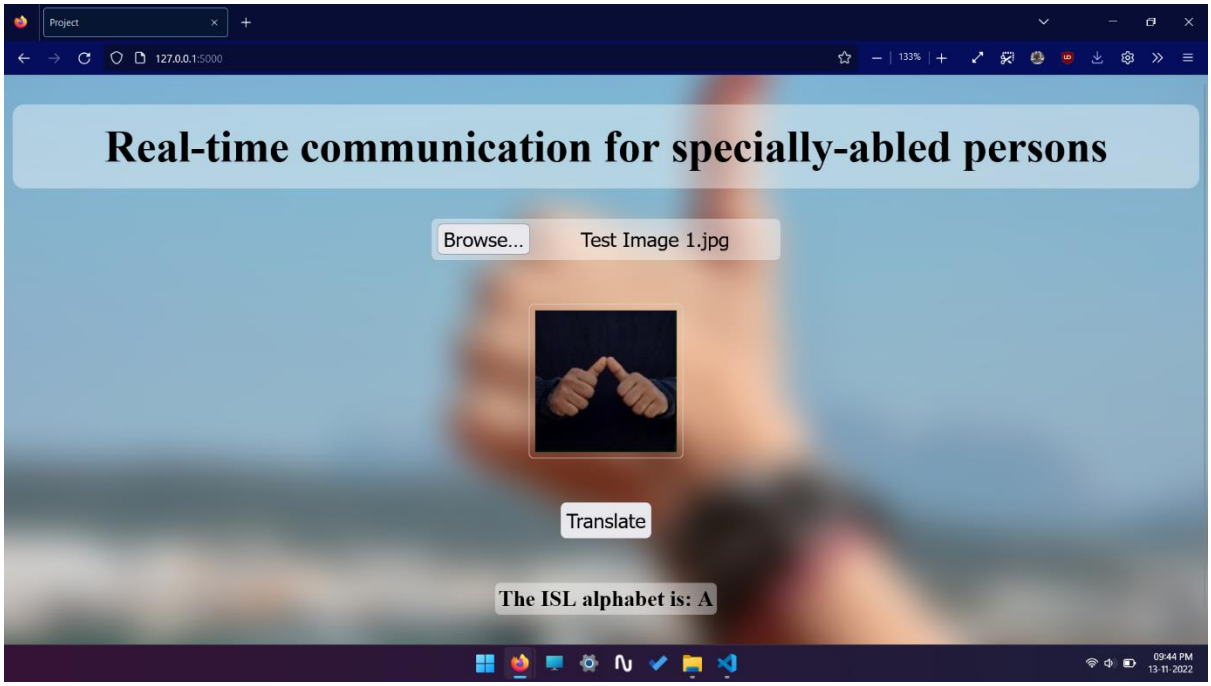
Python

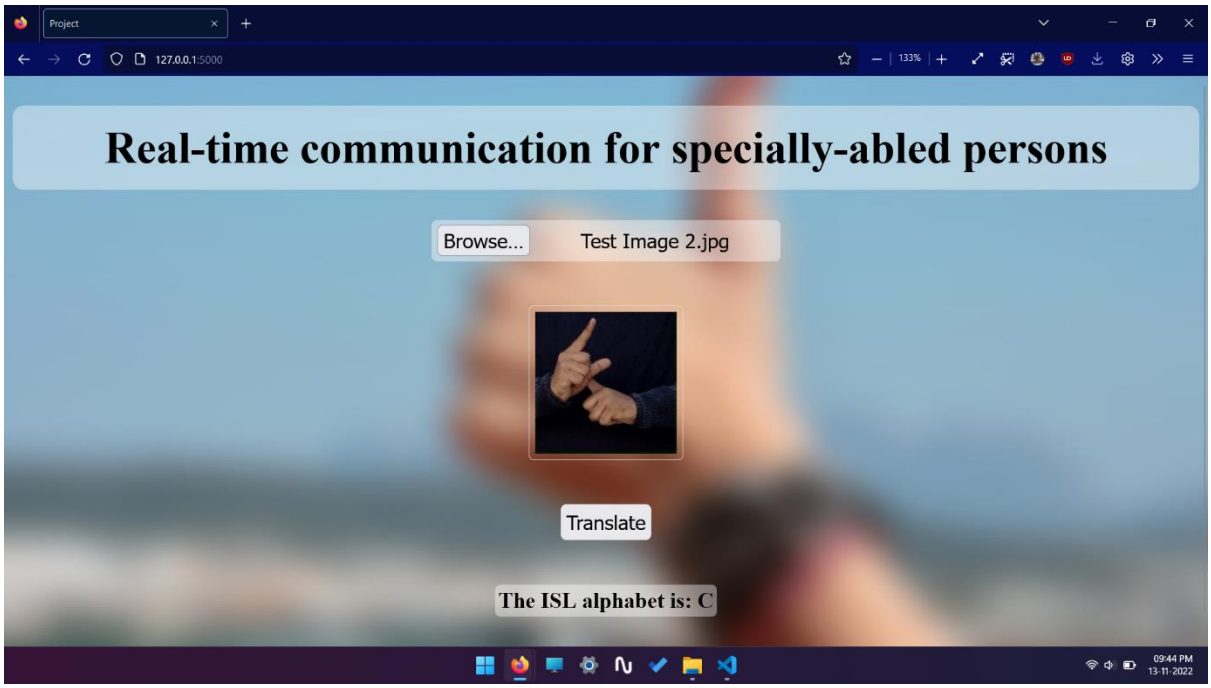
```
1/1 [==============================] - 0s 66ms/step
'C'
```

8.2 User Acceptance Testing:



Real-time usage output:





The created software (above) satisfies the user requirements and the purposes of our project listed in page no. 2.

9. RESULTS:

9.1 Performance Metrics:

RAM & CPU usage:

The software when run, only consumes:

- minimal RAM of around 1.8 to 7 MB [1]
- CPU usage of around 1.4% during translation [1][2]

[1] – Results are computed based on the average of running the software in multiple computers. Also, results may vary from one computer to another.

[2] – Tested on Ryzen 5 4500U (Hexa-core processor) with base clock 2.8 GHz.

Evaluation metrics:

The trained Convolutional Neural Network (CNN) model used in the software has an accuracy score of more than 95%.

10. ADVANTAGES & DISADVANTAGES:

Advantages:

- It lets the specially-abled persons communicate with normal person using their sign language.
- If used properly, the software can bridge the gap between specially-abled persons and normal persons.
- The application is scalable, i.e., its scope can be expanded to recognize digits, words, etc.

Disadvantages:

- Specially-abled persons need to be trained to work with the software.
- Since the trained model's accuracy is not 100%, sometimes there may be cases where the model may produce erroneous results.
- The input image needs to be of good quality for the model to classify correctly.

11. CONCLUSION:

- The project we developed can bridge the gap of communication between deaf-mute people and the normal people.
- The project can be expanded with several functionalities in future.
- The project also has a business potential which can be tapped.

12. FUTURE SCOPE:

This project has tremendous scope for future work. As this software includes components integrated into a Flask web application, components can be replaced or updated as per the changing requirements in the future.

Many new functionalities can be added like:

- Making the application predict words by combing multiple signs.
- Using NLP (Natural Language Processing) & ML (Machine Learning) to combine predicted words to create a meaningful sentence.
- Add many User Experience (UX) enhancing changes.
- Making the web application's UI (User Interface) look more appealing and user-friendly.

13. APPENDIX:

Source Code:

App.py:

app.py

E: > Computer Science and Engineering > Semester - 7 > IBM > Project Development Phase > Sprint 3 > Application Building > app.py > ...

```python
from flask import Flask, render_template, request, redirect, url_for
import backend

app = Flask(__name__)
path=''

@app.route('/', methods =["GET","POST"])
def index():
    global path
    if request.method == "POST":
        path = request.form.get("myInput")
        print(path)
        return render_template("translate.html", img_path=str(path), pred=str(backend.predi
    return render_template("index.html")


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

Backend.py

backend.py 2

E: > Computer Science and Engineering > Semester - 7 > IBM > Project Development Phase > Sprint 3 > Application Building > backend.py > ...

```python
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

model = load_model('trained_model.h5')

def get_prediction(num):
  out = {0:'A', 1:'B', 2:'C', 3:'D', 4:'E', 5:'F', 6:'G', 7:'H', 8:'I', 9:'J', 10:'K',
         11:'L', 12:'M', 13:'N', 14:'O', 15:'P', 16:'Q', 17:'R', 18:'S', 19:'T', 20:'U',
         22:'W', 23:'X', 24:'Y', 25:'Z'}
  return out[num]

def predict(img_src):
  img = image.load_img(img_src, target_size=(128,128), color_mode='grayscale')
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)
  prediction = np.argmax(model.predict(x))
  return get_prediction(prediction)

```

Index.html:

index.html

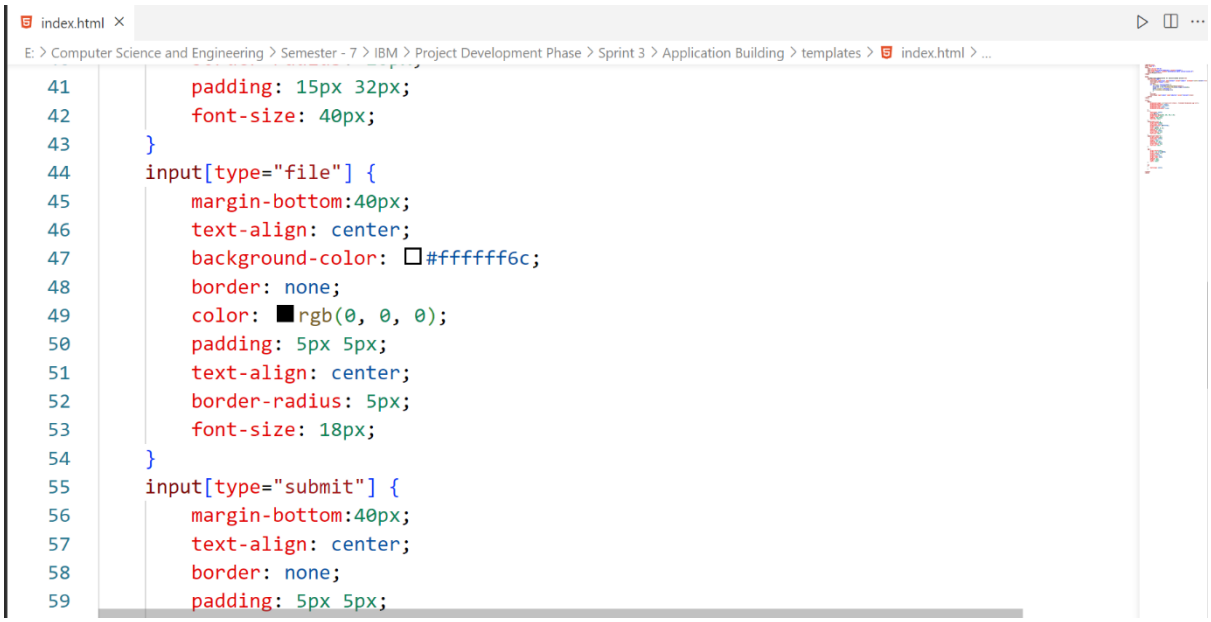
E: > Computer Science and Engineering > Semester - 7 > IBM > Project Development Phase > Sprint 3 > Application Building > templates > index.html > ...

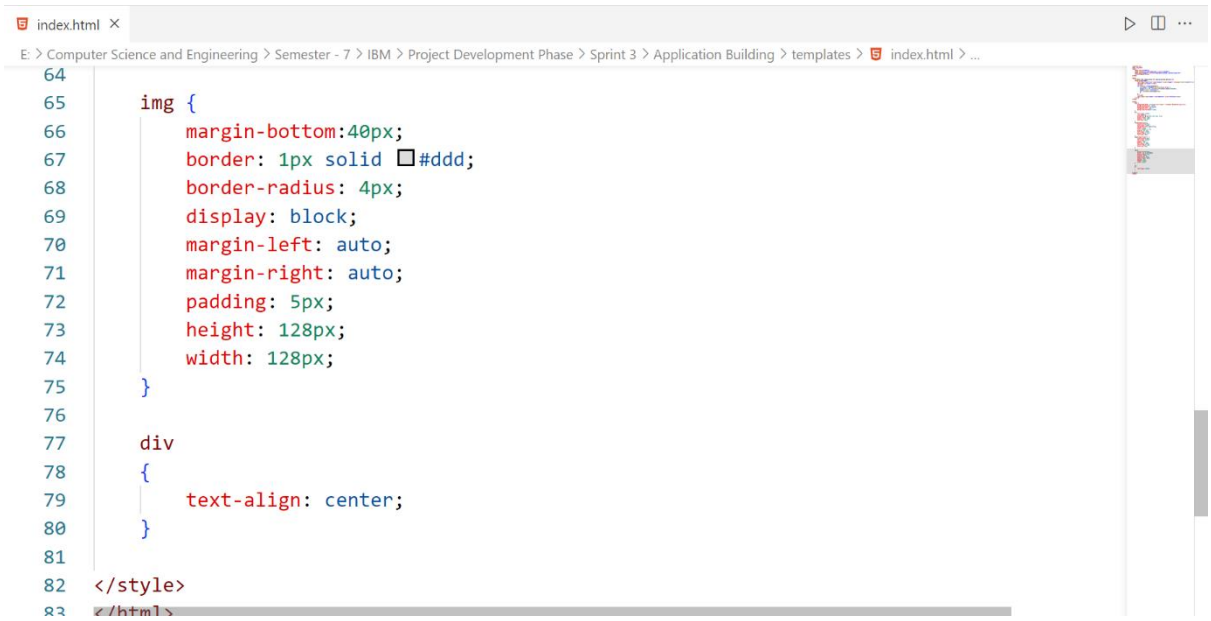
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Project</title>
</head>

<body>
    <h1>Real-time communication for specially-abled persons</h1>
    <form method="POST">
        <div><input type="file" name="myInput" accept="image/*" onchange="loadFile(event)"
        <div><img id="output"/></div>
        <script>
        var loadFile = function(event) {
            var output = document.getElementById('output');
            output.src = URL.createObjectURL(event.target.files[0]);
            output.onload = function() {
```

index.html

E: > Computer Science and Engineering > Semester - 7 > IBM > Project Development Phase > Sprint 3 > Application Building > templates > index.html > ...

```
            }
        };
        </script>
        <div><input type="submit" name="myButton" value="Translate"></div>
    </form>
</body>

<style>
    body {
        background-image: url("{{url_for('static', filename='Background.jpg')}}");
        background-position: center;
        background-repeat: no-repeat;
        background-size: cover;
        background-attachment: fixed;
    }
    h1 {
        text-align: center;
        color: black;
        background: rgba(255, 255, 255, 0.4);
```





GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-33649-1660224968

Project Demo Video Link:

https://drive.google.com/file/d/1dv6wbdUXDrhNVN4lRmOrV2bEdi_yu7Dg/view?usp=sharing