REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

Submitted By

Team Id: PNT2022TMID09766

Saroj Mandal	:	Team Leader[730919104102]
Phurpa Thelen Sherpa	:	Team Member[730919104076]
Shuvankar Sasmal	:	Team Member[730919104107]
Vibhakar Mishra	:	Team Member[730919104124]

CONTENTS

1. INTRODUCTION

1.1 Project Overview

Communication plays a significant role in making the world a better place. Communication creates bonding and relations among the people, whether persona, social, or political views. Most people communicate efficiently without any issues, but many cannot due to disability. They cannot hear or speak, which makes Earth a problematic place to live for them. Even simple basic tasks become difficult for them. Disability is an emotive human condition. It limits the individual to a certain level of performance. Being deaf and dumb pushes the subject to oblivion, highly introverted. In a world of inequality, this society needs empowerment. Harnessing technology to improve their welfare is necessary. In a tech era, no one should be limited due to his or her inability. The application of technology should create a platform or a world of equality despite the natural state of humans. On the other hand, technology is the most innovative thing on Earth for every time the clock ticks, researchers, software engineers, programmers, and information technology specialists are always coming up with bright ideas to provide convenience to everyone. This paper shows how artificial intelligence is being used to help people who are unable to do what most people do in their everyday lives. Aligned with communication, D-talk is a system that allows people who are unable to talk and hear be fully understood and for them to learn their language easier and also for the people that would interact and communicate with them. This system provides detailed hand gestures that show the interpretation at the bottom so that everyone can understand them. This research allows the readers to learn the system and what it can do to people who are struggling with what they are not capable of and will provide the technical terms on how the system works.

1.2 Purpose

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.

2. LITERATURE SURVEY

2.1 Existing problem

Some of the existing solutions for solving this problem are:

Technology

One of the easiest ways to communicate is through technology such as a smart phone or laptop. A deaf person can type out what they want to say and a person who is blind or has low vision can use a screen reader to read the text out loud. A blind person can also use voice recognition software to convert what they are saying in to text so that a person who is Deaf can then read it.

Interpreter

If a sign language interpreter is available, this facilitates easy communication if the person who is deaf is fluent in sign language. The deaf person and person who is blind can communicate with each other via the interpreter. The deaf person can use sign language and the interpreter can speak what has been said to the person who is blind and then translate anything spoken by the blind person into sign language for the deaf person. *Just Speaking*

Depending on the deaf person's level of hearing loss, they may be able to communicate with a blind person who is using speech. For example, a deaf person may have enough residual hearing (with or without the use of an assistive hearing device such as a hearing aid) to be able to decipher the speech of the person who is blind or has low vision. However, this is often not the most effective form of communication, as it is very dependent on the individual circumstances of both people and their environment (for example, some places may have too much background noise).

2.2 References

1) **A Signer Independent Sign Language Recognition with Coarticulation Elimination from Live Videos: an Indian Scenario** P.K. Athira, C.J. Sruthi, A. Lijiya (2019) **Advantage:**

   Economical can be implemented with a mobile camera which makes it very user-friendly

   Disadvantage:

   Not efficient under cluttered backgrounds and different illumination conditions

2) **A Deep Learning based Indian Sign Language Recognition System** Sruthi C. J and Lijiya A (2019) **Advantage:**

   Training accuracy of 99.93% and with testing and validation accuracy of 98.64%.

   Disadvantage:

   Facial expression and context analysis are the other part not included

3) **Hand Gesture Recognition for Sign Language Using 3DCNN** Muneer AlHammadi, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaiman, Mohamed A. Bencherif, And Mohamed Amine Mekhtiche (2020)

Advantage:

The proposed approaches were compared with six other state-of-the art methods from the literature. They outperformed four of these methods and showed comparable performance to the other two.

Disadvantage:

Does not work for a live video feed.

2.3 Problem Statement Definition

Communication is the only medium by which we can share our thoughts or convey the message but communications between deafmute 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.

Problem:

The boy has difficulty in hearing. He uses sign language to communicate with others. But he can’t able to communicate with normal people who don’t understand sign language.

Solution:

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, the system enhances the user friendly experience.

Problem:

Karupan is a dumb by birth. He uses sign language to communicate with others. But he can’t able to communicate with normal people who don’t understand sign language.

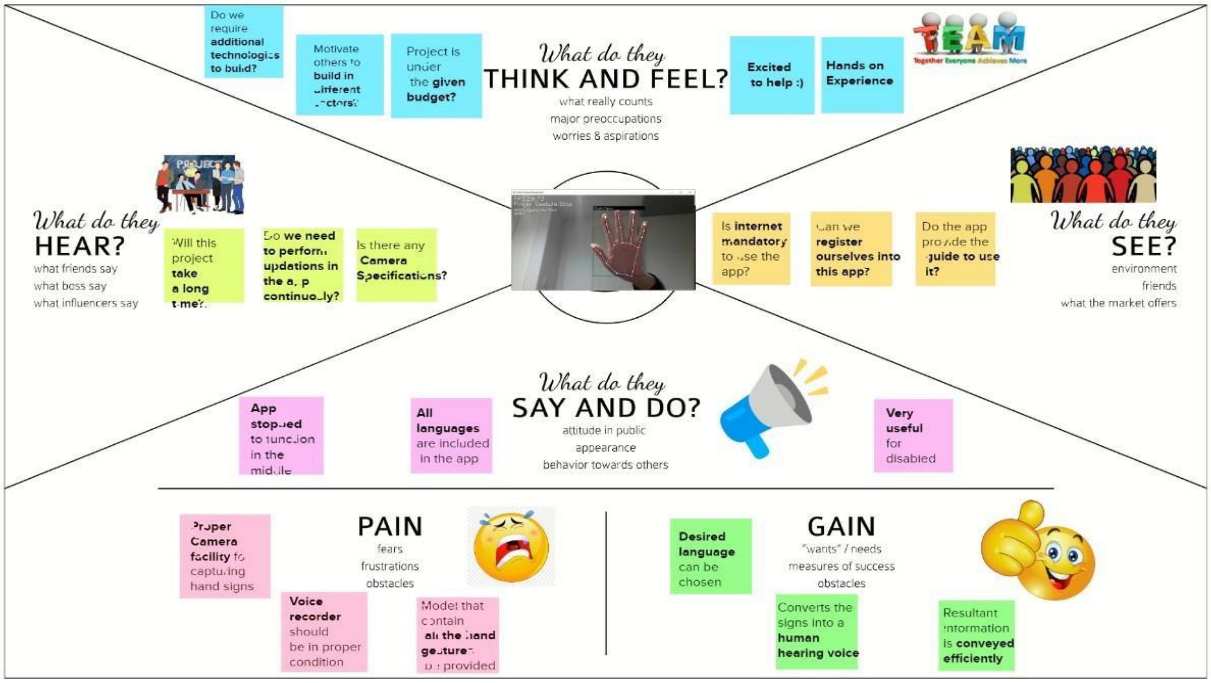
Solution:
To create a app for understanding sign language and convert into

Speech signal as output for normal people.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Build Empathy and keep your focus on the user by putting yourself in their shoes.



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

Do we require additional technologies to build?

Motivate others to build in different factors?

Project is under the given budget?

Excited to help :)

Hands on Experience

What do they HEAR?
what friends say
what boss say
what influencers say

Will this project take a long time?

Do we need to perform updations in the app continuously?

Is there any Camera Specifications?

What do they SEE?
environment
friends
what the market offers

Is internet mandatory to use the app?

Can we register ourselves into this app?

Do the app provide the guide to use it?

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

App stopped to function in the middle

All languages are included in the app

Very useful for disabled

PAIN
fears
frustrations
obstacles

Proper Camera facility to capturing hand signs

Voice recorder should be in proper condition

Model that contain all the hand gestures to provided

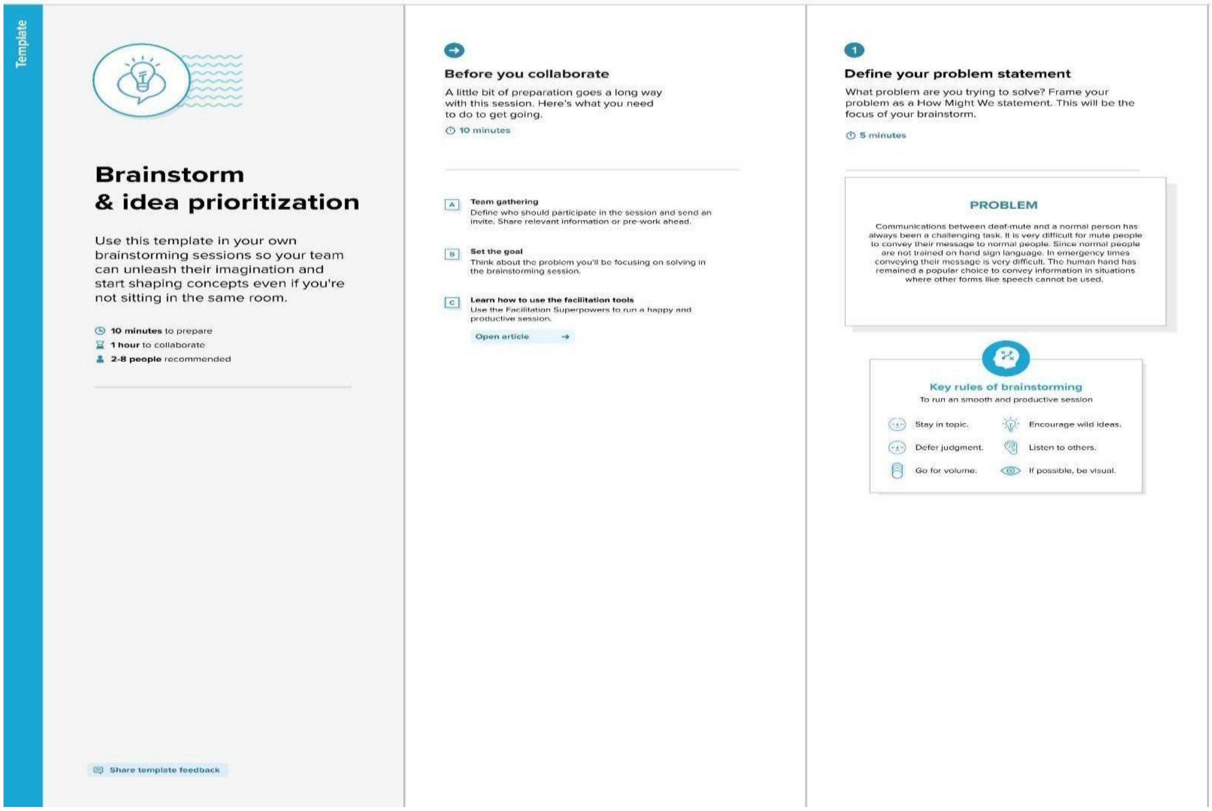
GAIN
"wants" / needs
measures of success
obstacles

Desired language can be chosen

Converts the signs into a human hearing voice

Resultant information is conveyed efficiently

3.2 Ideation & Brainstorming



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

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.

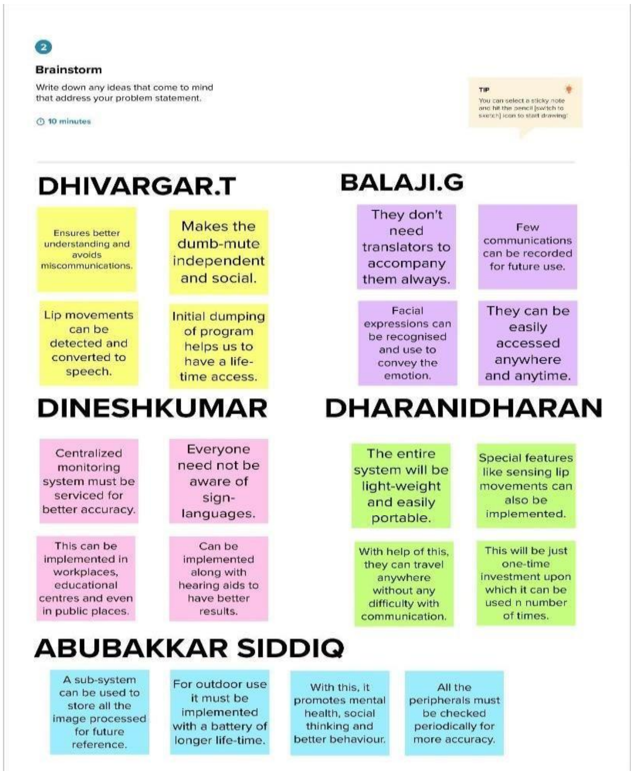
5 minutes

PROBLEM

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.

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

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

DHIVARGAR.T

Ensures better understanding and avoids miscommunications.

Makes the dumb-mute independent and social.

Lip movements can be detected and converted to speech.

Initial dumping of program helps us to have a life-time access.

BALAJI.G

They don't need translators to accompany them always.

Few communications can be recorded for future use.

Facial expressions can be recognised and use to convey the emotion.

They can be easily accessed anywhere and anytime.

DINESHKUMAR

Centralized monitoring system must be serviced for better accuracy.

Everyone need not be aware of sign-languages.

This can be implemented in workplaces, educational centres and even in public places.

Can be implemented along with hearing aids to have better results.

DHARANIDHARAN

The entire system will be light-weight and easily portable.

Special features like sensing lip movements can also be implemented.

With help of this, they can travel anywhere without any difficulty with communication.

This will be just one-time investment upon which it can be used n number of times.

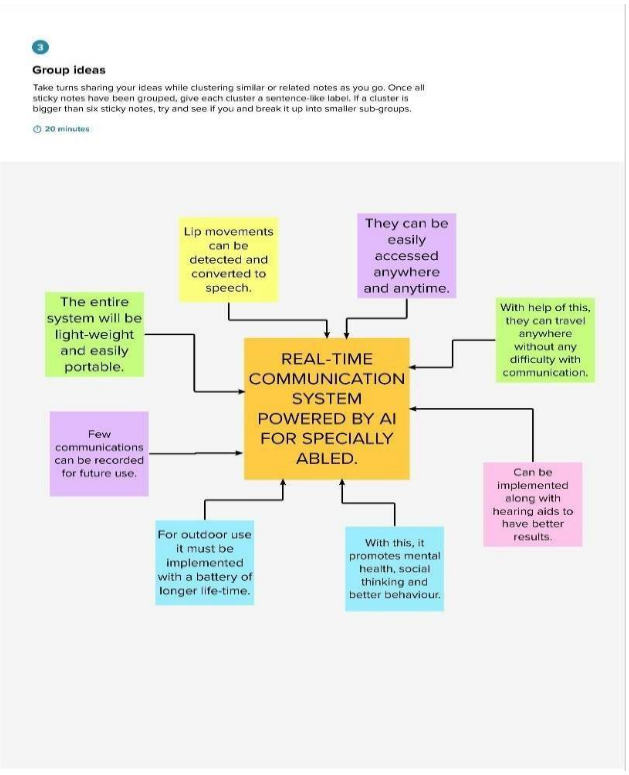
ABUBAKKAR SIDDIQ

A sub-system can be used to store all the image processed for future reference.

For outdoor use it must be implemented with a battery of longer life-time.

With this, it promotes mental health, social thinking and better behaviour.

All the peripherals must be checked periodically for more accuracy.



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

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED.

Lip movements can be detected and converted to speech.

They can be easily accessed anywhere and anytime.

The entire system will be light-weight and easily portable.

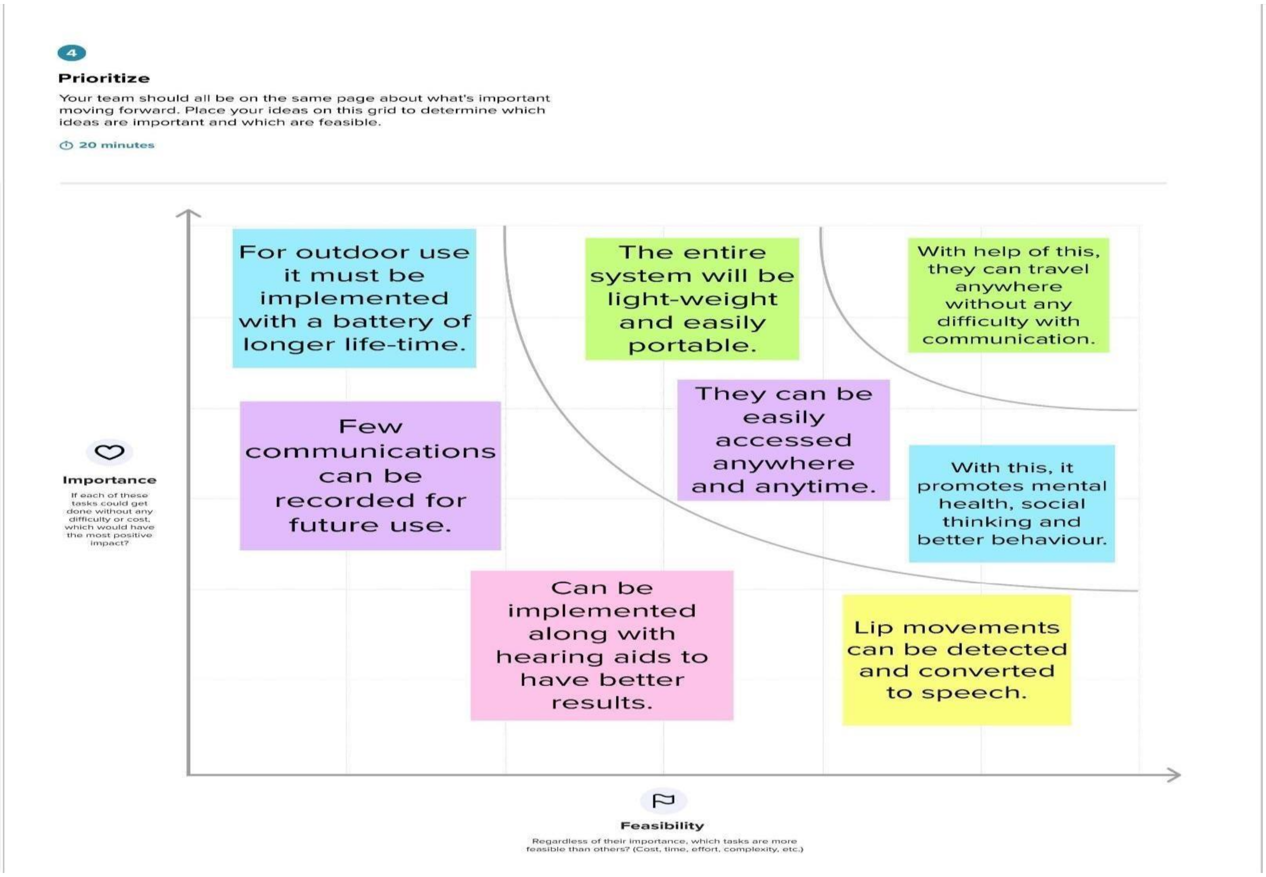
With help of this, they can travel anywhere without any difficulty with communication.

Few communications can be recorded for future use.

Can be implemented along with hearing aids to have better results.

For outdoor use it must be implemented with a battery of longer life-time.

With this, it promotes mental health, social thinking and better behaviour.



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

For outdoor use it must be implemented with a battery of longer life-time.

The entire system will be light-weight and easily portable.

With help of this, they can travel anywhere without any difficulty with communication.

Few communications can be recorded for future use.

They can be easily accessed anywhere and anytime.

With this, it promotes mental health, social thinking and better behaviour.

Can be implemented along with hearing aids to have better results.

Lip movements can be detected and converted to speech.

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.)

3.3 Proposed Solution

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Everyone is not convenient with language used in the application • Some people cannot understand English we can convert into their convenient language • They are facing difficulties in understanding the language used in the system.
2.	Idea / Solution description	• Even sign language can also be translated to text message in our application using CNN. • Text to sign language convertor uses Stanford Parser text processing and JA Signing for the signing avatar • Can change the language using google language translator tool so that people can use the application based on their specialized language • Producing a model which can recognize Finger-spelling based hand gestures in order to form a complete word by combining each gesture • By using this application both specially abled and normal people can translate their messages to others easily

3.	Novelty / Uniqueness	• Convenient language can be changed using the google language translator tool • Normal text can also be translated into sign language
4.	Social Impact / Customer Satisfaction	• The Main aim of the project is to build an application that helps the especially abled people to communicate with others easily • The deaf and dumb people can easilytranslate their sign language into a human hearing voice • The normal people can also easily translate their voice into a sign language using this application
5.	Business Model (Revenue Model)	• We can generate revenue by offering subscription-based applications to the people • Users who have got subscription can change the language accordingly

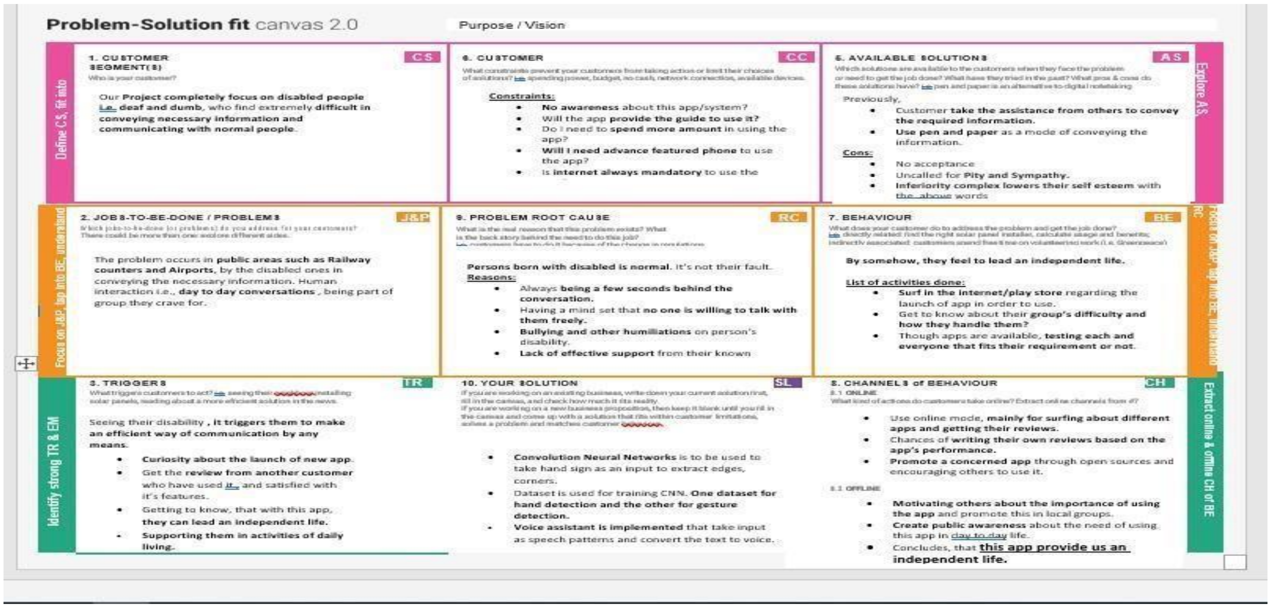
6.	Scalability of the Solution	• Even if the number of users increase the system will perform well • Need to pay attention to the application and to be responsive to the changes as fast as possible

3.4 Problem Solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why **Purpose:**

- ✈ Solve complex problems in a way that fits the state of your customers.
- ✈ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- ✈ Sharpen your communication and marketing strategy with the right triggers and messaging.
- ✈ Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
- ✈ **Understand the existing situation in order to improve it for your target group.**

Template:



Problem-Solution fit canvas 2.0

Purpose / Vision

Define C S, fit into

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Our **Project completely focus on disabled people i.e. deaf and dumb**, who find extremely **difficult in conveying necessary information and communicating with normal people**.

6. CUSTOMER CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Constraints:

- **No awareness** about this app/system?
- Will the app **provide the guide to use it?**
- Do I need to **spend more amount** in using the app?
- **Will I need advance featured phone** to use the app?
- Is **internet always mandatory** to use the

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Previously,

- Customer **take the assistance from others to convey the required information.**
- **Use pen and paper** as a mode of conveying the information.

Cons:

- No acceptance
- Uncalled for **Pity and Sympathy.**
- **Inferiority complex lowers their self esteem** with the above words

Explore AS,

Focus on J&P, tap into BE, understand

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

The problem occurs in **public areas such as Railway counters and Airports**, by the disabled ones in conveying the necessary information. Human interaction i.e., **day to day conversations** , being part of group they crave for.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.

Persons born with disabled is normal. It's not their fault.

Reasons:

- Always **being a few seconds behind the conversation.**
- Having a mind set that **no one is willing to talk with them freely.**
- **Bullying and other humiliations** on person's disability.
- **Lack of effective support** from their known

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

By somehow, they feel to lead an independent life.

List of activities done:

- **Surf in the internet/play store** regarding the launch of app in order to use.
- Get to know about their **group's difficulty and how they handle them?**
- Though apps are available, **testing each and everyone that fits their requirement or not.**

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Seeing their disability **, it triggers them to make an efficient way of communication by any means.**

- **Curiosity about the launch of new app.**
- Get the **review from another customer** who have used it, and satisfied with it's features.
- Getting to know, that with this app, **they can lead an independent life.**
- **Supporting them in activities of daily living.**

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

- **Convolution Neural Networks** is to be used to take hand sign as an input to extract edges, corners.
- Dataset is used for training CNN. **One dataset for hand detection and the other for gesture detection.**
- **Voice assistant is implemented** that take input as speech patterns and convert the text to voice.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

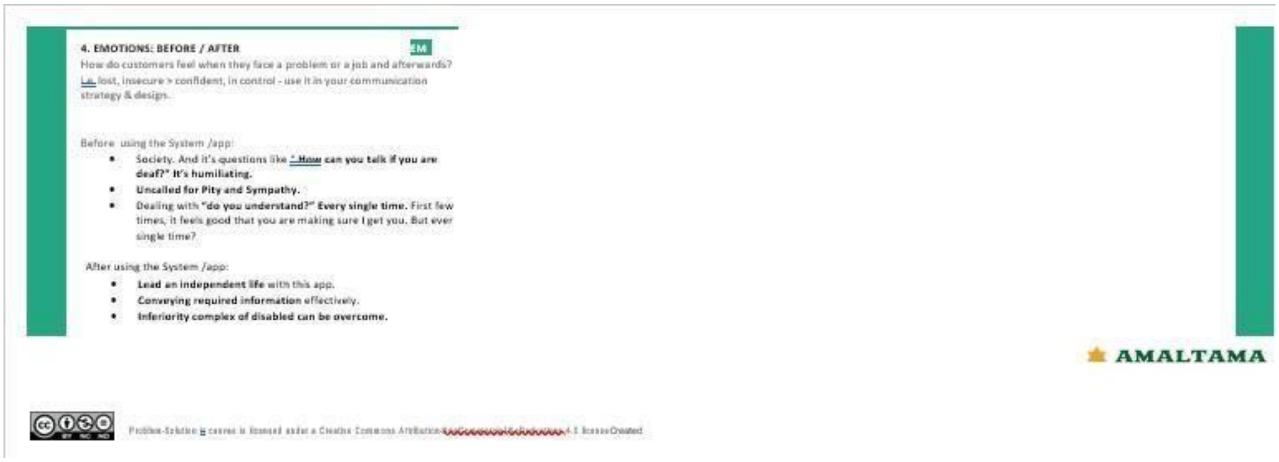
What kind of actions do customers take online? Extract online channels from #7

- Use online mode, **mainly for surfing about different apps and getting their reviews.**
- Chances of **writing their own reviews based on the app's performance.**
- **Promote a concerned app** through open sources and encouraging others to use it.

8.2 OFFLINE

- **Motivating others about the importance of using the app** and promote this in local groups.
- **Create public awareness** about the need of using this app in day to day life.
- Concludes, that **this app provide us an independent life.**

Extract online & offline CH of BE



4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before using the System /app:

- Society. And it's questions like **" How can you talk if you are deaf?" It's humiliating.**
- **Uncalled for Pity and Sympathy.**
- Dealing with **"do you understand?" Every single time.** First few times, it feels good that you are making sure I get you. But ever single time?

After using the System /app:

- **Lead an independent life** with this app.
- **Conveying required information** effectively.
- **Inferiority complex of disabled can be overcome.**

AMALTAMA

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	• Registration through Web UI/ E-Mail ID. • Authentication via OTP.
FR-2	User Confirmation	□ Confirmation via mail.
FR-3	System	• Desktop/ Mobile with good resolution camera. • Provides system access to capture images/ video and other relevant data.
FR-4	Text conversion	Converts the Sign language into a text using Convolutional Neural Network (CNN) Model.
FR-5	Sentence Translation	To create sentence(s) by recognizing the signs and pauses in the input video stream.

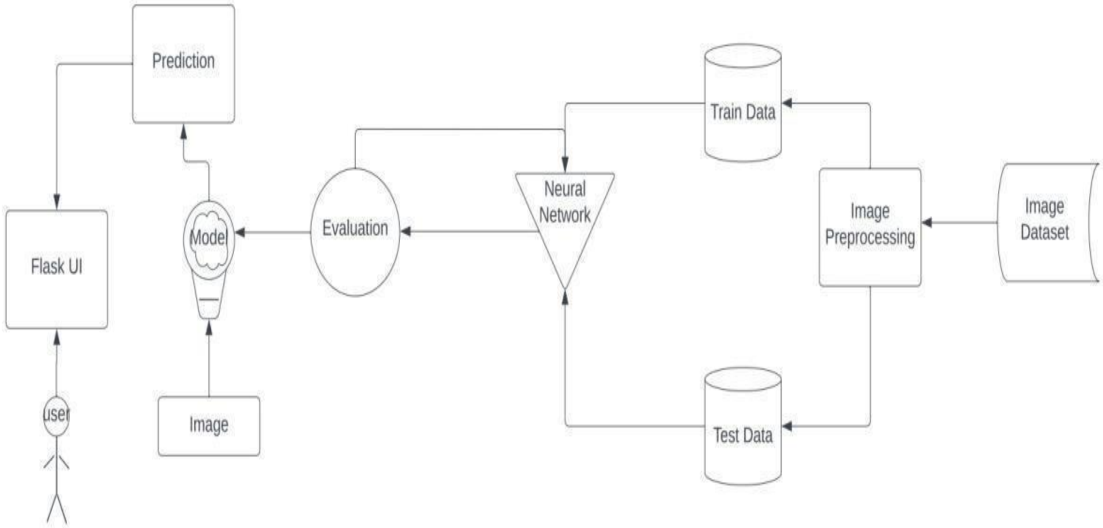
4.2 Non-Functional Requirement

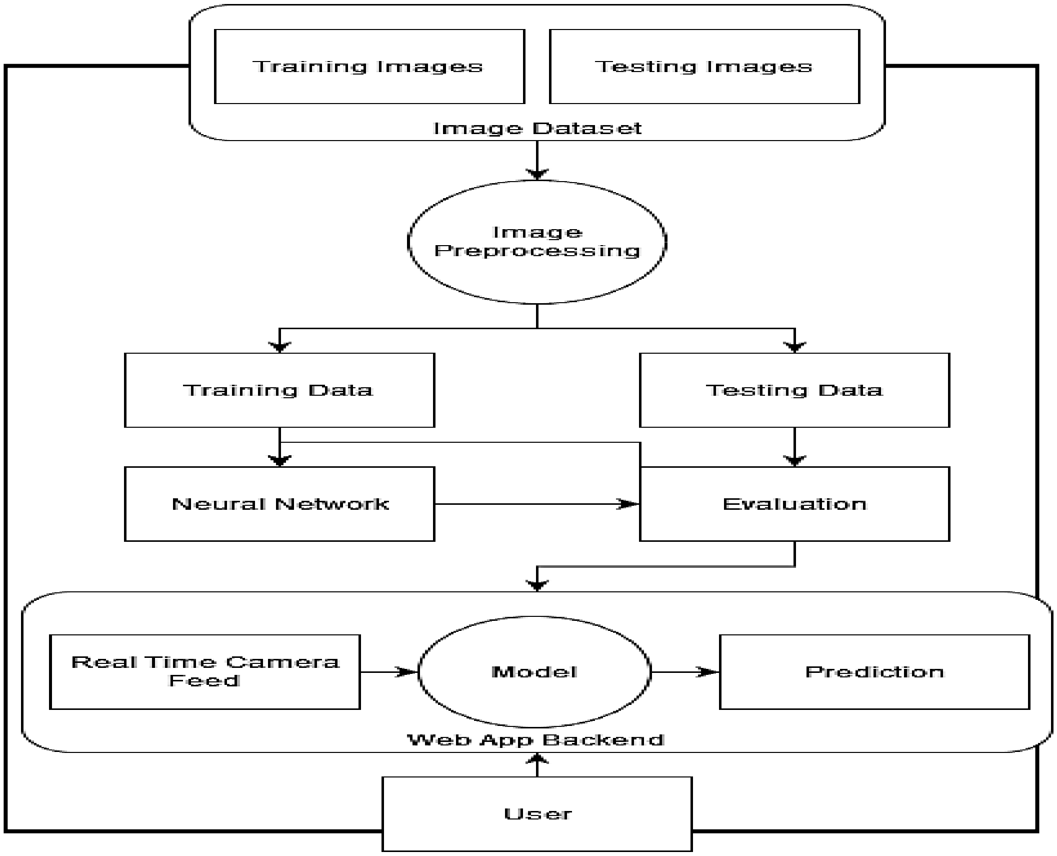
NFR No.	Non-Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)

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 considerable time so that 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

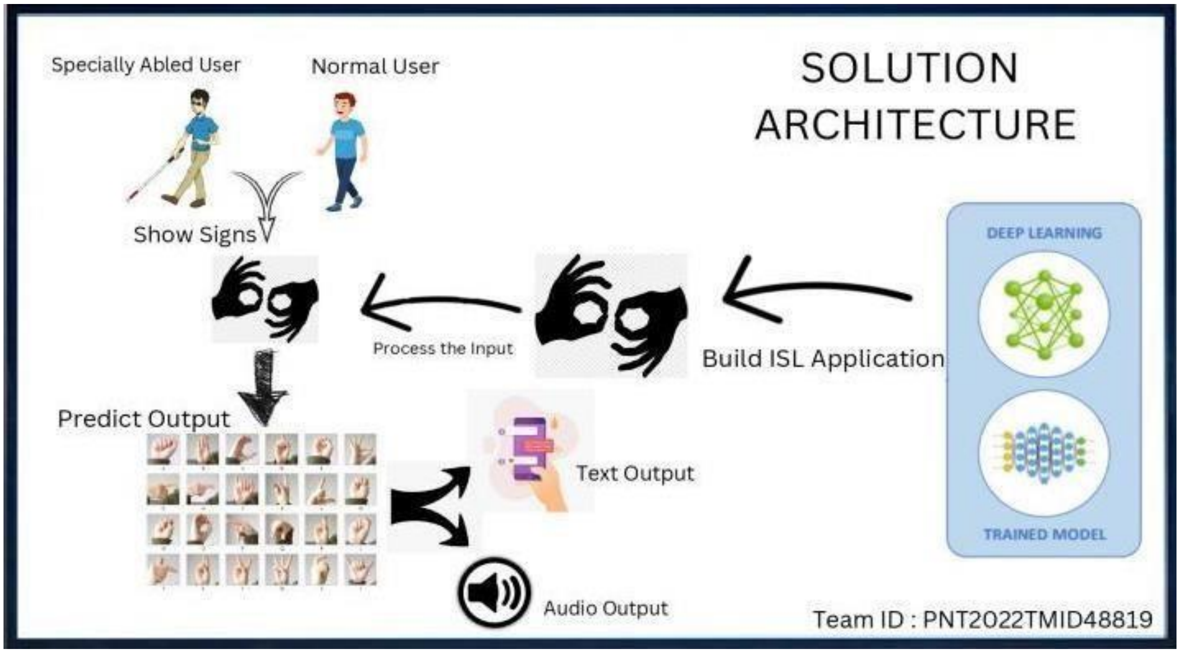
5.1 Dataflow Diagram



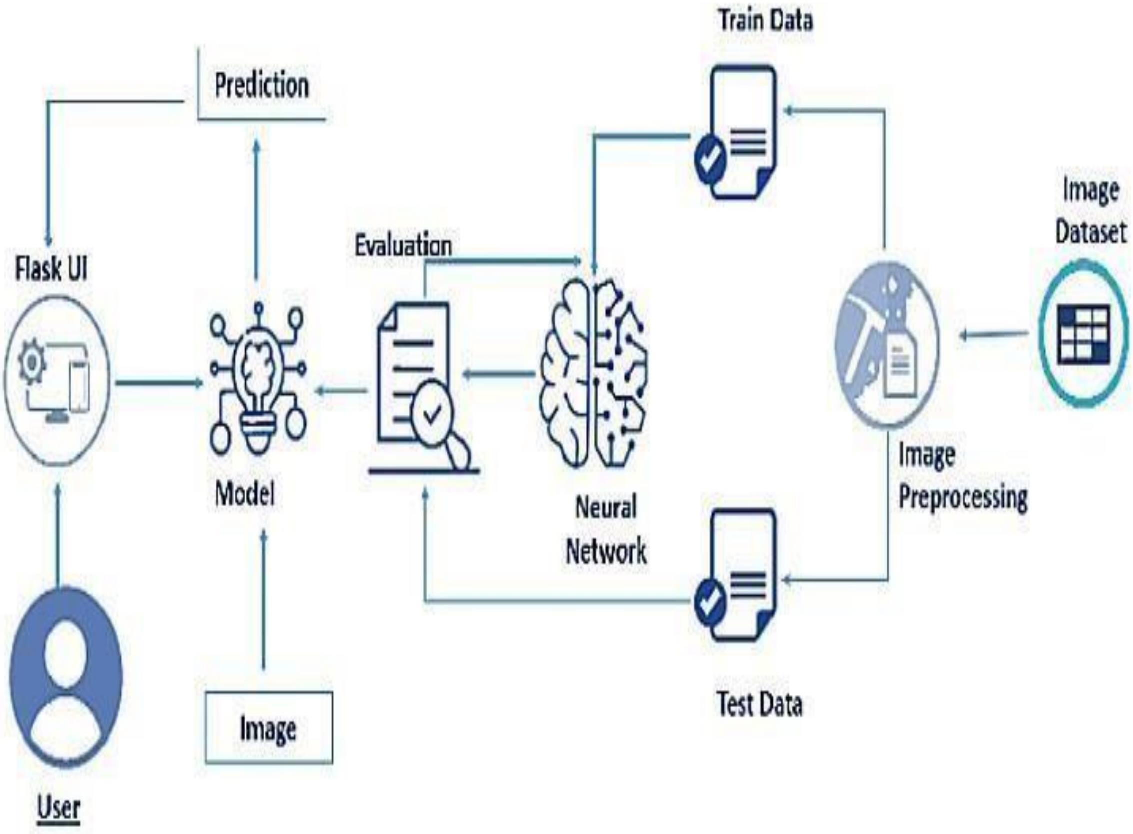


5.2 Solution & Technical Architecture

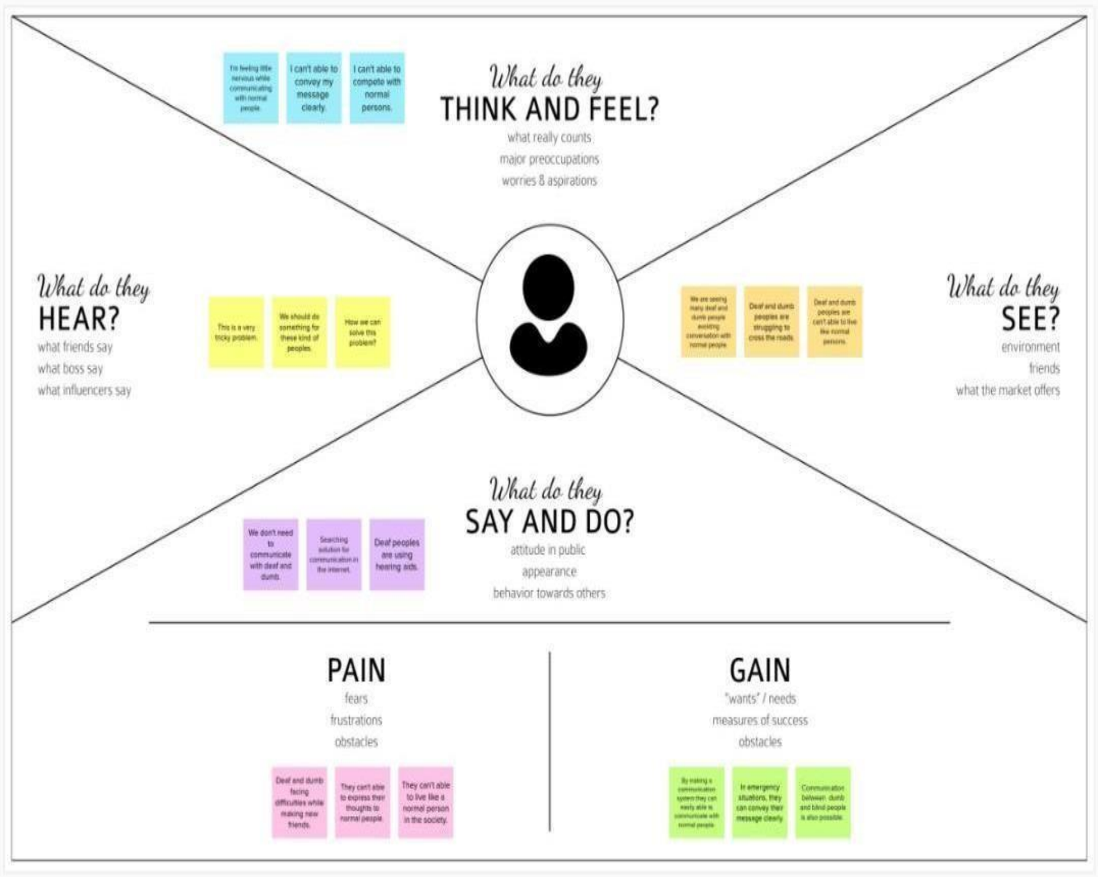
Solution architecture



Technical Architecture



5.3 User Stories:


I'm feeling little nervous while communicating with normal people
I can't able to convey my message clearly.
I can't able to compete with normal persons.
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries 8 aspirations
What do they
HEAR?
what friends say
what boss say
what influencers say
This is a very tricky problem.
We should do something for these kind of peoples.
How we can solve this problem?
We are seeing many deaf and dumb people avoiding conversation with normal people.
Deaf and dumb peoples are struggling to cross the roads.
Deaf and dumb peoples are can't able to live like normal persons.
What do they
SEE?
environment
friends
what the market offers
We don't need to communicate with deaf and dumb.
Searching solution for communication in the internet.
Deaf peoples are using hearing aids.
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
Deaf and dumb facing difficulties while making new friends.
They can't able to express their thoughts to normal people.
They can't able to live like a normal person in the society.
GAIN
"wants" / needs
measures of success
obstacles
By making a communication system they can easily able to communicate with normal people
In emergency situations, they can convey their message clearly
Communication between dumb and blind people is also possible.

User Type	Functional Reaquirement(Epic)	User Story Number	User Story/Task	Acceptance criteria
Customer (People who cannot hear)	Covert sign Language into text	USN-1	As a user,I can Open camera in the appa and record my signs to be convereted into text	I can communicate With normal people effectively
		USN-2	As a user,I can upload my previous sign gestures to communicate faster	I can have a list of frequently used signs to make for fast reference
	Dashboard	USN-3	Buttons to record the signs,to convert in realtime and other buttons should be available in the right places	All features must be easily accessible

		USN-4	Emergency calls Must be available so that I can press a button in times of emergency to get the attention of others	I can feel safe because of the Emergency Button Which can get me help.
Customer(People who can hear talk)	Convert sign language into text	USN-5	As a user, I can open back camera in the app and record the specially abled people’s signs to be converted into text	I can understand the mode of communication of specially-abled people effectively
		USN-6	As a user, I can open a Text-pad that is available in the app, so that the deaf people can see the message I need to convey	I can convey my message to them effectively
Administrator	Integrate appliation with trained model	USN-7	As an admin, I should be able to integrate the AI model into the application and maintain the application	I can give best experience to app users

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Use the below template to create product backlog and planning

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	Communi-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Saroj mandal, Shuvankar Sasmal
Sprint-1	Registration	Communi-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Vibhakar Mishra, Purpa Thelen Sherpa
Sprint-2	Registration	Communi-3	As a user, I can register for the application through Phone number	2	Medium	Shuvankar Sasmal
Sprint-2	User Interface	Communi-4	Professional responsible for user requirements and needs	2	Medium	Purpa Thelen Sherpa
Sprint-3	Login	Communi-5	As a user, I can log into the application by entering email & password	1	High	Saroj mandal ,Vibhakar Mishra
Sprint-3	Dashboard	Communi-6	As a user, I must receive any updates or pop ups in my dashboard	2	High	Purpa Thelen Sherpa, Saroj mandal
Sprint-4	Details	Communi-7	As a user, I should get notification about the progress and any updates via email or sms	1	Medium	Saroj mandal
Sprint-4	Privacy	Communi-8	The developed application should be secure for the users	2	High	Shuvankar Sasmal, Vibhakar Mishra

6.2 Sprint Delivery Schedule

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	10	6 Days	25 Oct 2022	31 Oct 2022	10	31 Oct 2022
Sprint-2	10	6 Days	1 Nov 2022	06 Nov 2022	10	06 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	13 Nov 2022	10	14 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	22 Nov 2022

6.3.REPORTS FROM JIRA

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

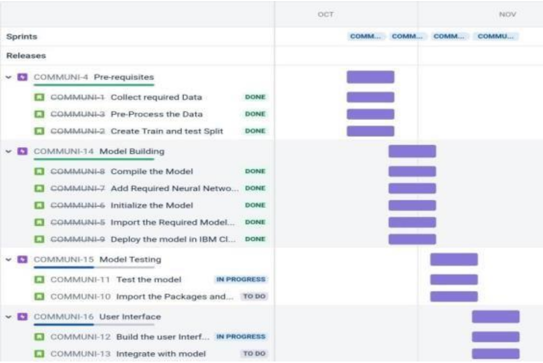
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

AV = 6/10 = 0.6 **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.

Sprint Burnout Chart:



Sprints COMM... COMM... COMM... COMMU...

Releases

COMMUNI-4 Pre-requisites

~~COMMUNI-1~~ Collect required Data DONE

~~COMMUNI-3~~ Pre-Process the Data DONE

~~COMMUNI-2~~ Create Train and test Split DONE

COMMUNI-14 Model Building

~~COMMUNI-8~~ Compile the Model DONE

~~COMMUNI-7~~ Add Required Neural Netwo... DONE

~~COMMUNI-6~~ Initialize the Model DONE

~~COMMUNI-5~~ Import the Required Model... DONE

~~COMMUNI-9~~ Deploy the model in IBM Cl... DONE

COMMUNI-15 Model Testing

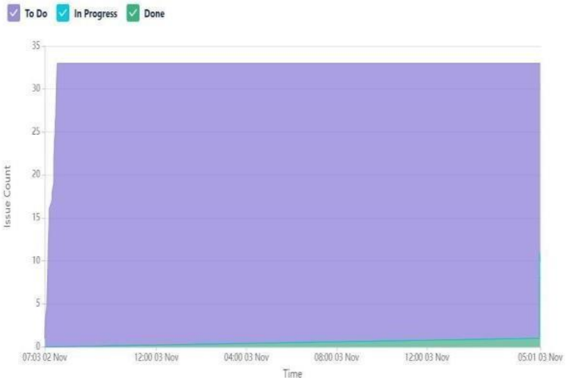
COMMUNI-11 Test the model IN PROGRESS

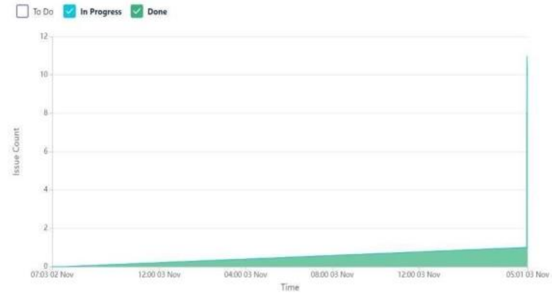
COMMUNI-10 Import the Packages and... TO DO

COMMUNI-16 User Interface

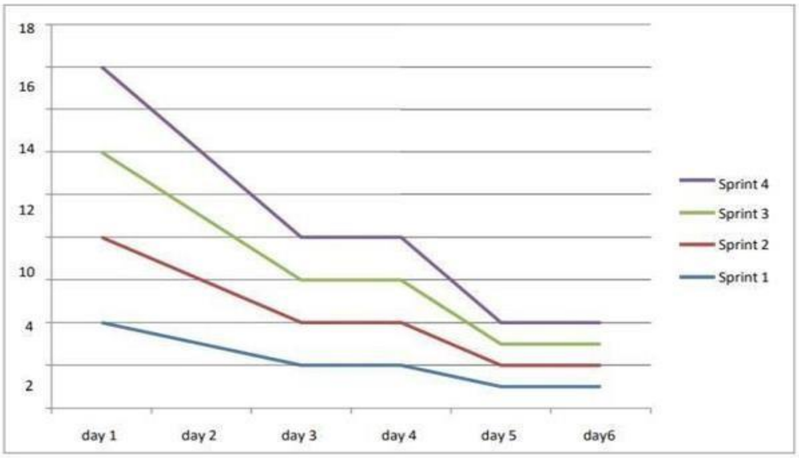
COMMUNI-12 Build the user Interf... IN PROGRESS

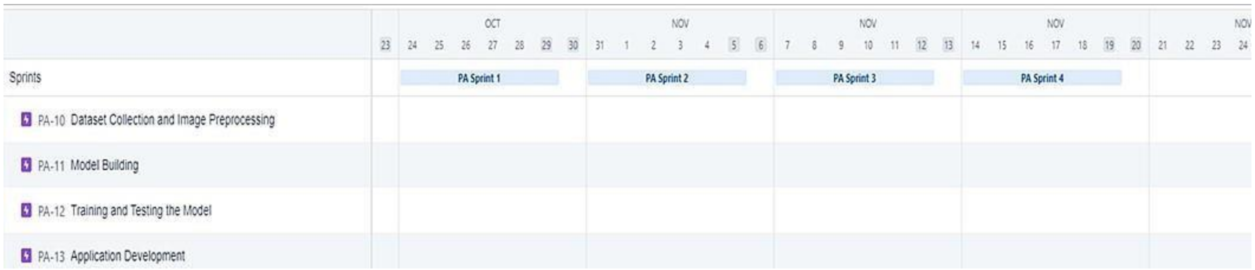
COMMUNI-13 Integrate with model TO DO





Sprint Schedule Chart:





Sprints PA Sprint 1 PA Sprint 2 PA Sprint 3 PA Sprint 4

PA-10 Dataset Collection and Image Preprocessing

PA-11 Model Building

PA-12 Training and Testing the Model

PA-13 Application Development

7. CODING & SOLUTIONING

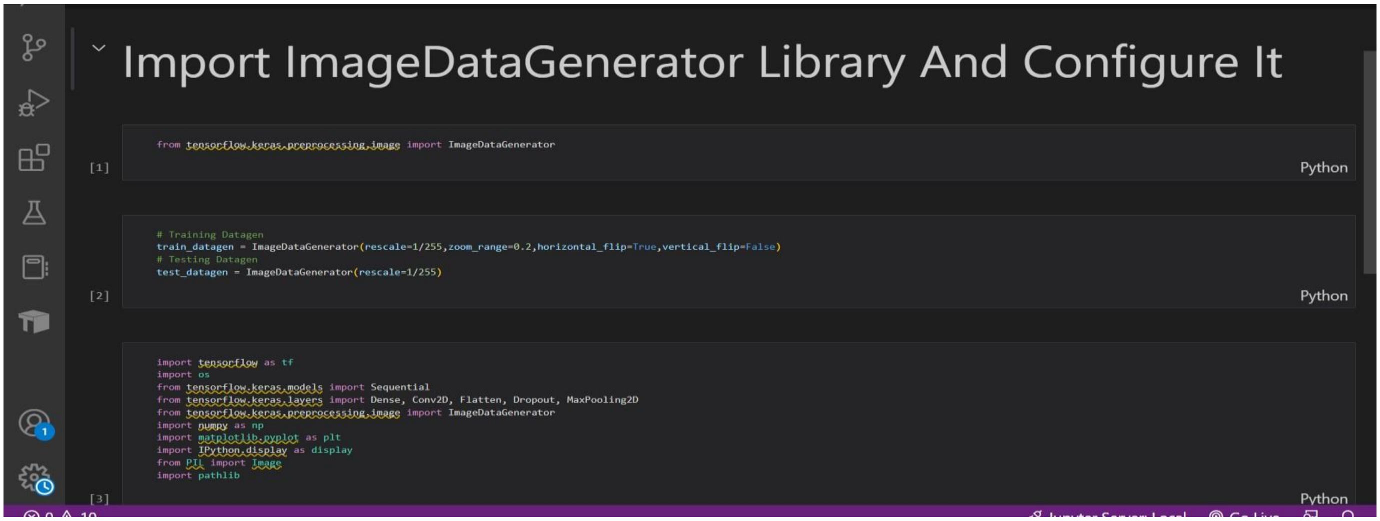
(Explain the features added in the project along with code)

7.1 *Feature 1*

IMAGE PREPROCESSING

➢ Image pre-processing includes zooming, shearing, flipping to increase the robustness of the model after it is built. Keras package is used for pre-processing images.

- Importing Image Data Generator Library to create an instance for which include shearing, rescale, zooming, etc to make the model robust with different types of images.



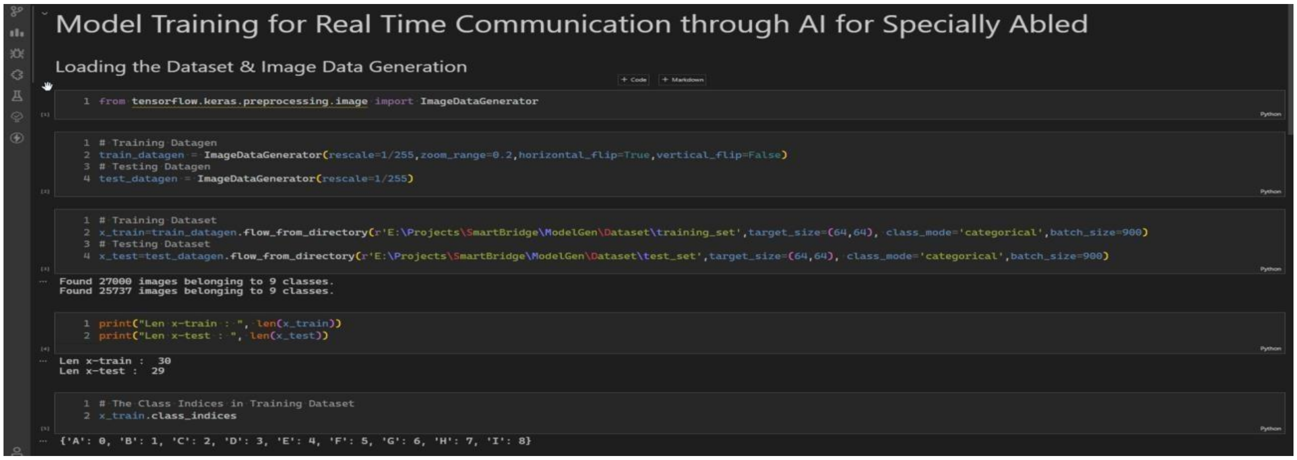
Import ImageDataGenerator Library And Configure It

```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```python
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```python
import tensorflow as tf
import os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import IPython.display as display
from PIL import Image
import pathlib
```

<u>Model Training</u>



Model Training for Real Time Communication through AI for Specially Abled

Loading the Dataset & Image Data Generation

```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```python
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```python
# Training Dataset
x_train=train_datagen.flow_from_directory(r'E:\Projects\SmartBridge\ModelGen\Dataset\training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

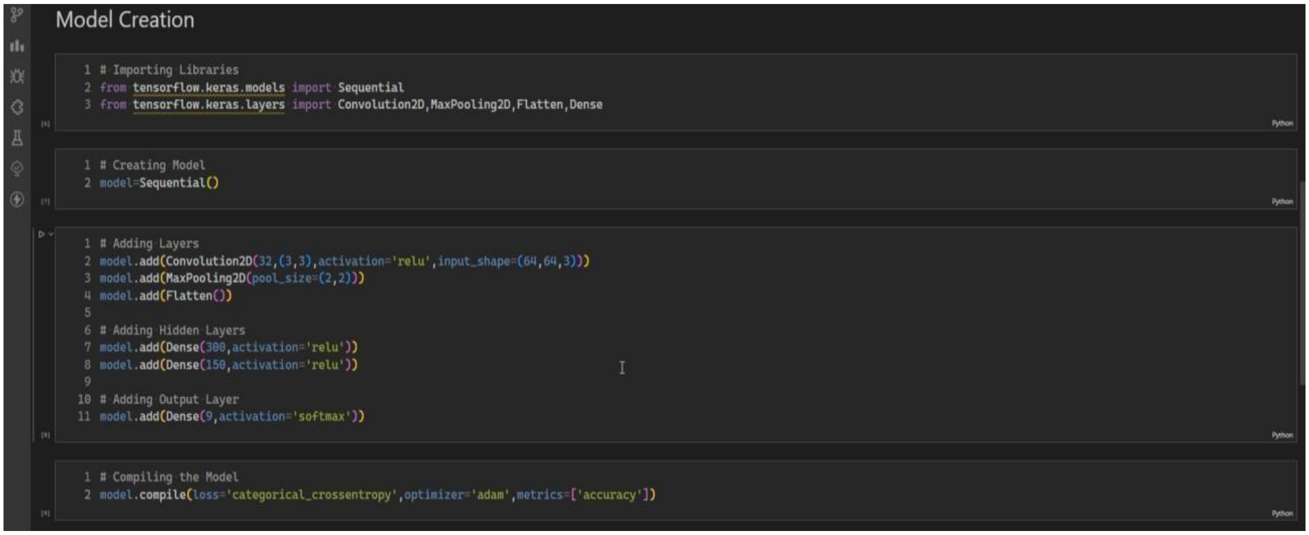
```
Found 27000 images belonging to 9 classes.
Found 25737 images belonging to 9 classes.
```

```python
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train :  30
Len x-test :  29
```

```python
# The Class Indices in Training Dataset
x_train.class_indices
```

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```



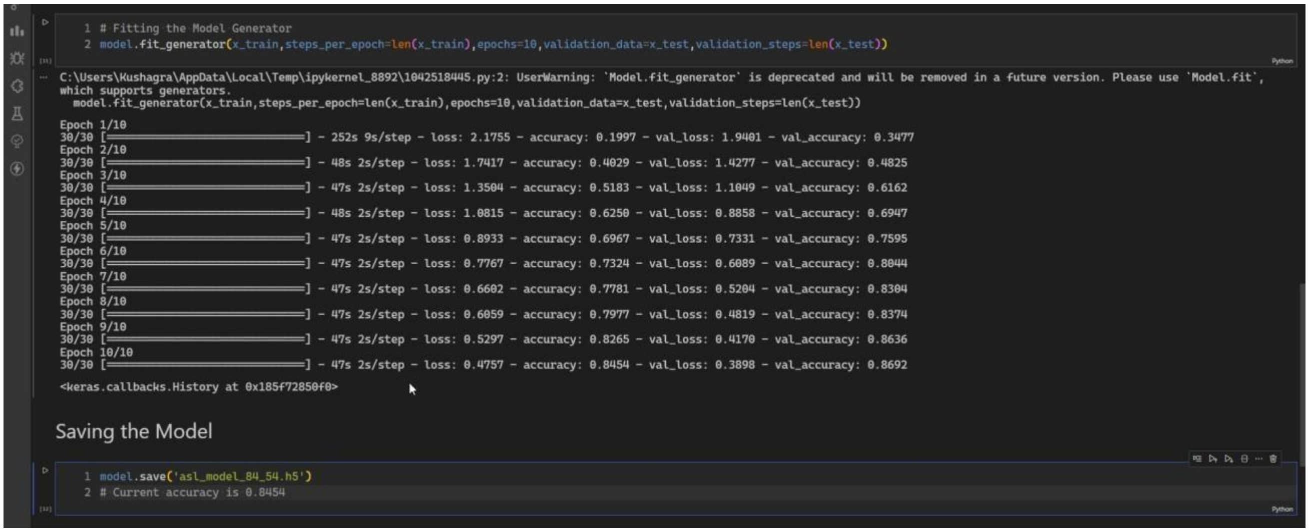
Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

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

```
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

# Adding Hidden Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))

# Adding Output Layer
model.add(Dense(9,activation='softmax'))
```

```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```



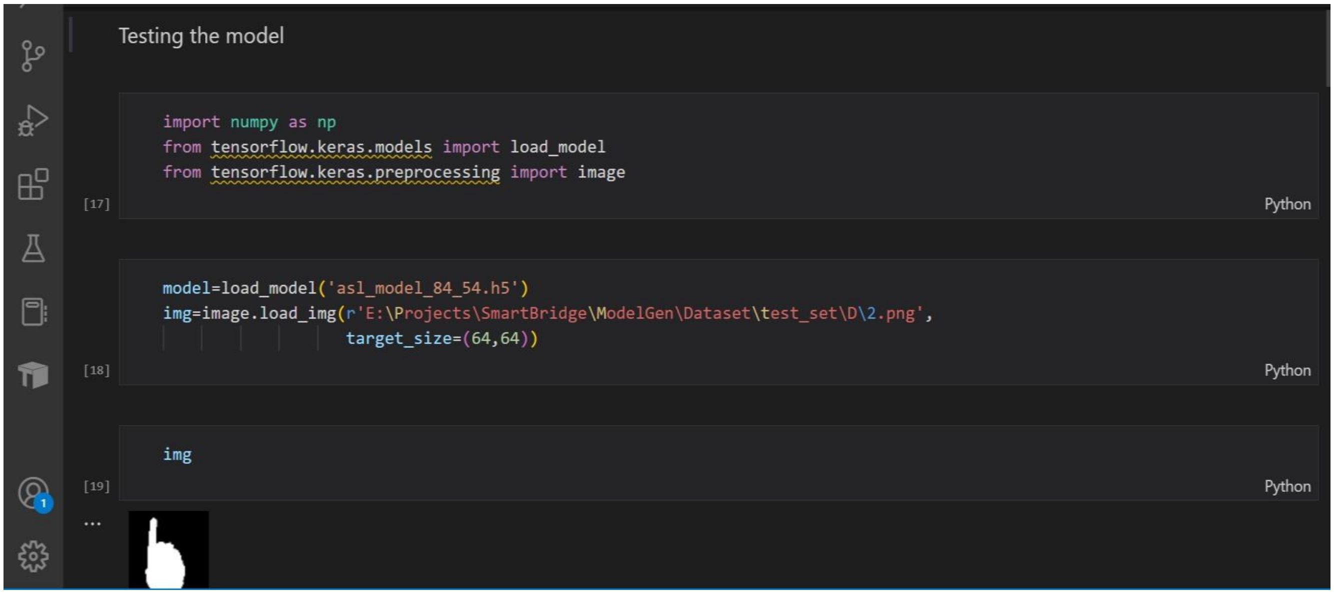
```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

```
C:\Users\Kushagra\AppData\Local\Temp\ipykernel_8892\1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`,
which supports generators.
  model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))

Epoch 1/10
30/30 [==============================] - 252s 9s/step - loss: 2.1755 - accuracy: 0.1997 - val_loss: 1.9401 - val_accuracy: 0.3477
Epoch 2/10
30/30 [==============================] - 48s 2s/step - loss: 1.7417 - accuracy: 0.4029 - val_loss: 1.4277 - val_accuracy: 0.4825
Epoch 3/10
30/30 [==============================] - 47s 2s/step - loss: 1.3504 - accuracy: 0.5183 - val_loss: 1.1049 - val_accuracy: 0.6162
Epoch 4/10
30/30 [==============================] - 48s 2s/step - loss: 1.0815 - accuracy: 0.6250 - val_loss: 0.8858 - val_accuracy: 0.6947
Epoch 5/10
30/30 [==============================] - 47s 2s/step - loss: 0.8933 - accuracy: 0.6967 - val_loss: 0.7331 - val_accuracy: 0.7595
Epoch 6/10
30/30 [==============================] - 47s 2s/step - loss: 0.7767 - accuracy: 0.7324 - val_loss: 0.6089 - val_accuracy: 0.8044
Epoch 7/10
30/30 [==============================] - 47s 2s/step - loss: 0.6602 - accuracy: 0.7781 - val_loss: 0.5204 - val_accuracy: 0.8304
Epoch 8/10
30/30 [==============================] - 47s 2s/step - loss: 0.6059 - accuracy: 0.7977 - val_loss: 0.4819 - val_accuracy: 0.8374
Epoch 9/10
30/30 [==============================] - 47s 2s/step - loss: 0.5297 - accuracy: 0.8265 - val_loss: 0.4170 - val_accuracy: 0.8636
Epoch 10/10
30/30 [==============================] - 47s 2s/step - loss: 0.4757 - accuracy: 0.8454 - val_loss: 0.3898 - val_accuracy: 0.8692

<keras.callbacks.History at 0x185f72850f0>
```

Saving the Model

```
model.save('asl_model_84_54.h5')
# Current accuracy is 0.8454
```

Model Testing

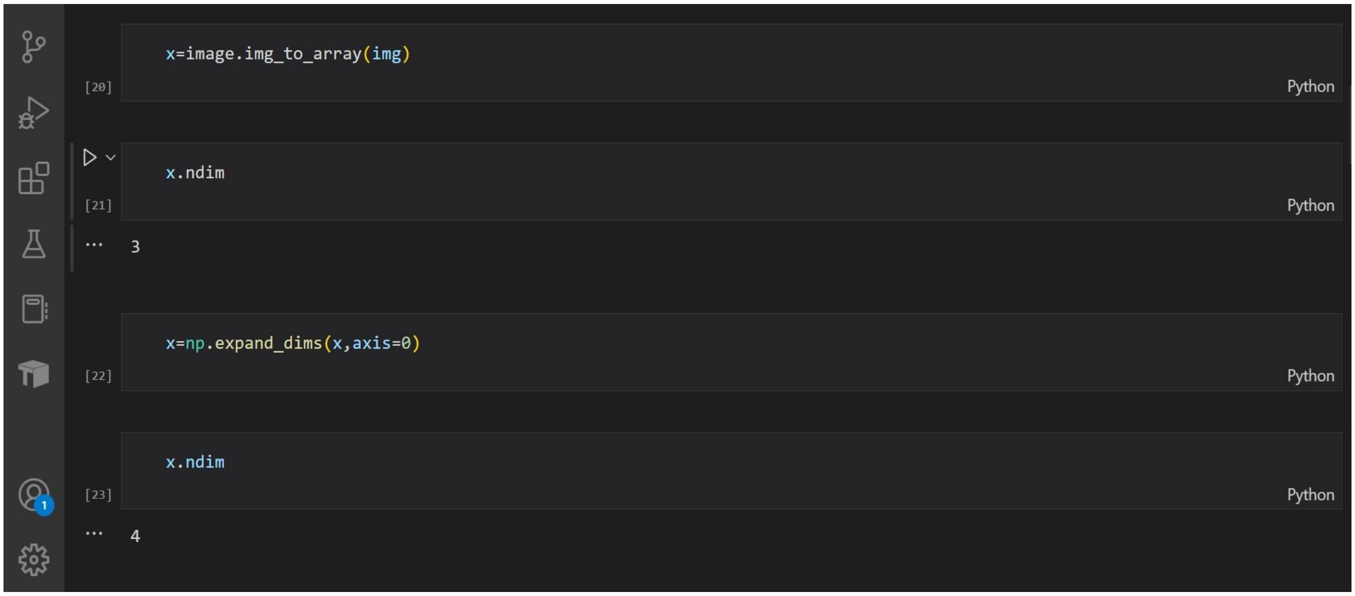


Testing the model

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

```
model=load_model('asl_model_84_54.h5')
img=image.load_img(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set\D\2.png',
                   target_size=(64,64))
```

```
img
```



```
[20]  x=image.img_to_array(img)
```

```
[21]  x.ndim
```

```
...   3
```

```
[22]  x=np.expand_dims(x,axis=0)
```

```
[23]  x.ndim
```

```
...   4
```

```
[20]  x=image.img_to_array(img)
```

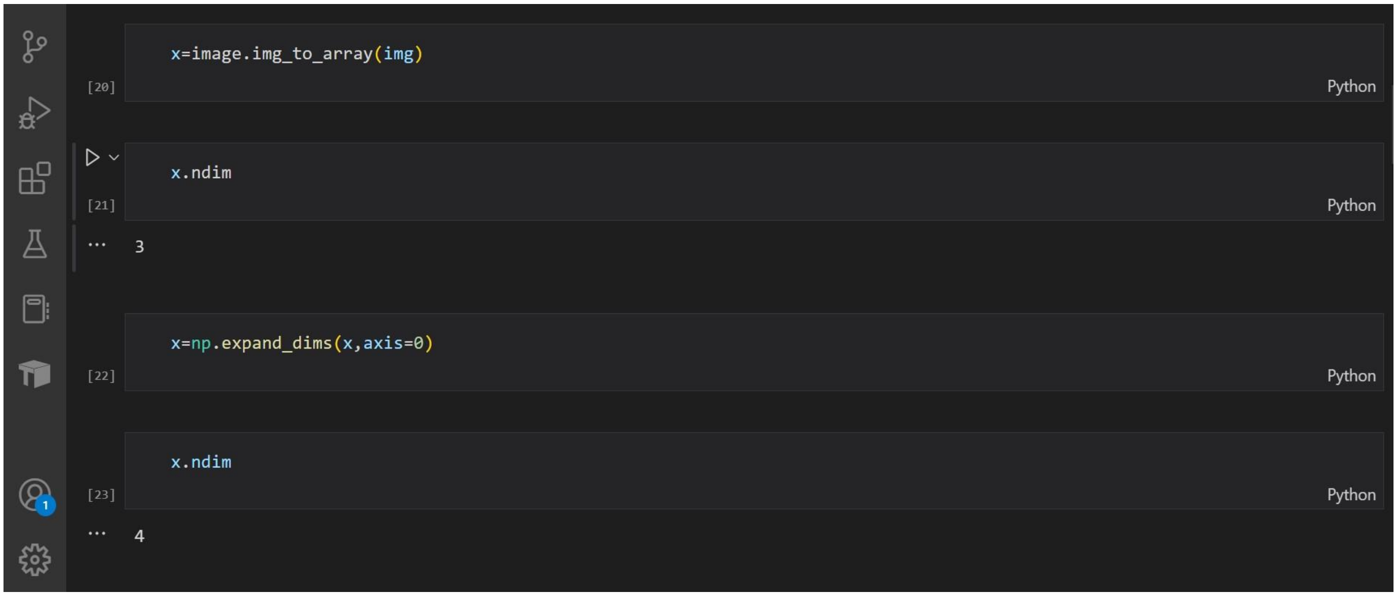
```
[21]  x.ndim
```

```
...   3
```

```
[22]  x=np.expand_dims(x,axis=0)
```

```
[23]  x.ndim
```

```
...   4
```



```
x=image.img_to_array(img)
```

```
x.ndim
```

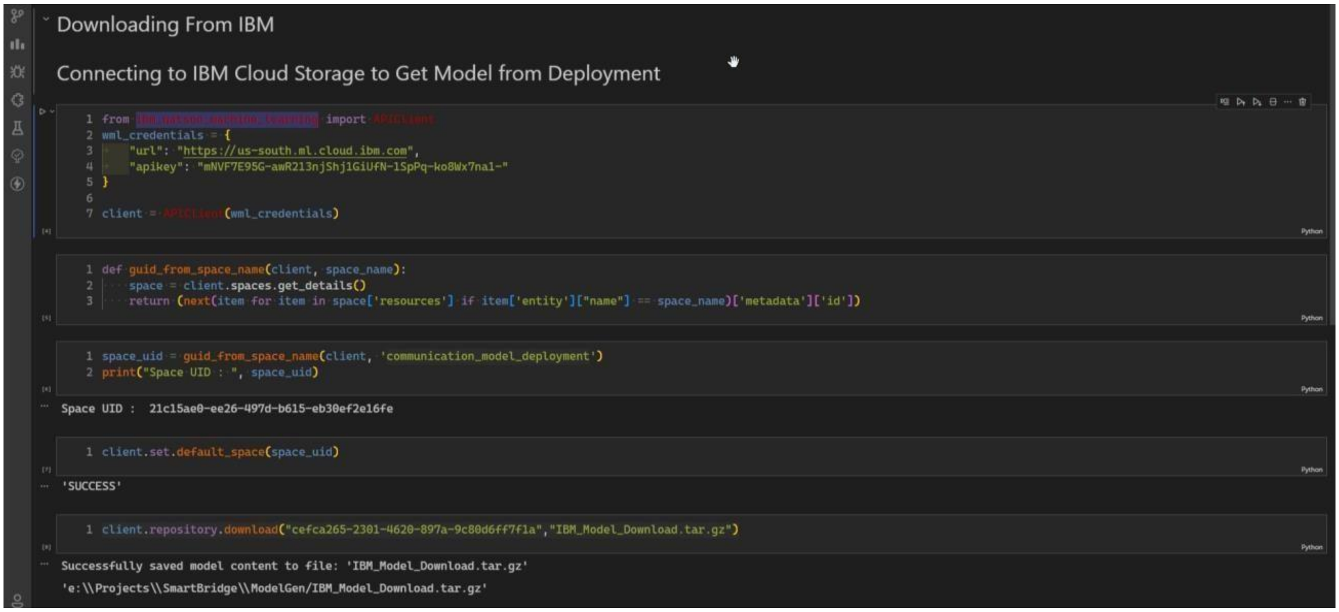
```
3
```

```
x=np.expand_dims(x,axis=0)
```

```
x.ndim
```

```
4
```

IBM Model Training And download



Downloading From IBM

Connecting to IBM Cloud Storage to Get Model from Deployment

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "mNVF7E95G-awR213njShj1GiUfN-1SpPq-ko8Wx7na1-"
}

client = APIClient(wml_credentials)
```

```
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return (next(item for item in space['resources'] if item['entity']["name"] == space_name)['metadata']['id'])
```

```
space_uid = guid_from_space_name(client, 'communication_model_deployment')
print("Space UID : ", space_uid)
```

```
Space UID :  21c15ae0-ee26-497d-b615-eb30ef2e16fe
```

```
client.set.default_space(space_uid)
```

```
'SUCCESS'
```

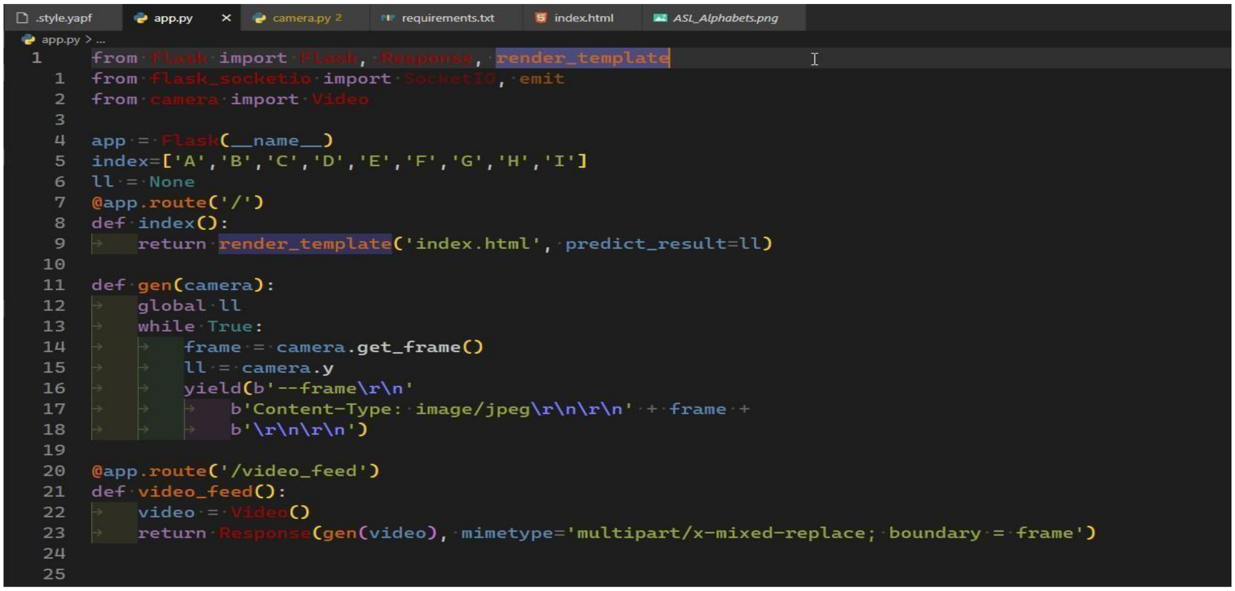
```
client.repository.download("cefca265-2301-4620-897a-9c80d6ff7f1a","IBM_Model_Download.tar.gz")
```

```
Successfully saved model content to file: 'IBM_Model_Download.tar.gz'
'e:\\Projects\\SmartBridge\\ModelGen/IBM_Model_Download.tar.gz'
```

7.2 Feature 2

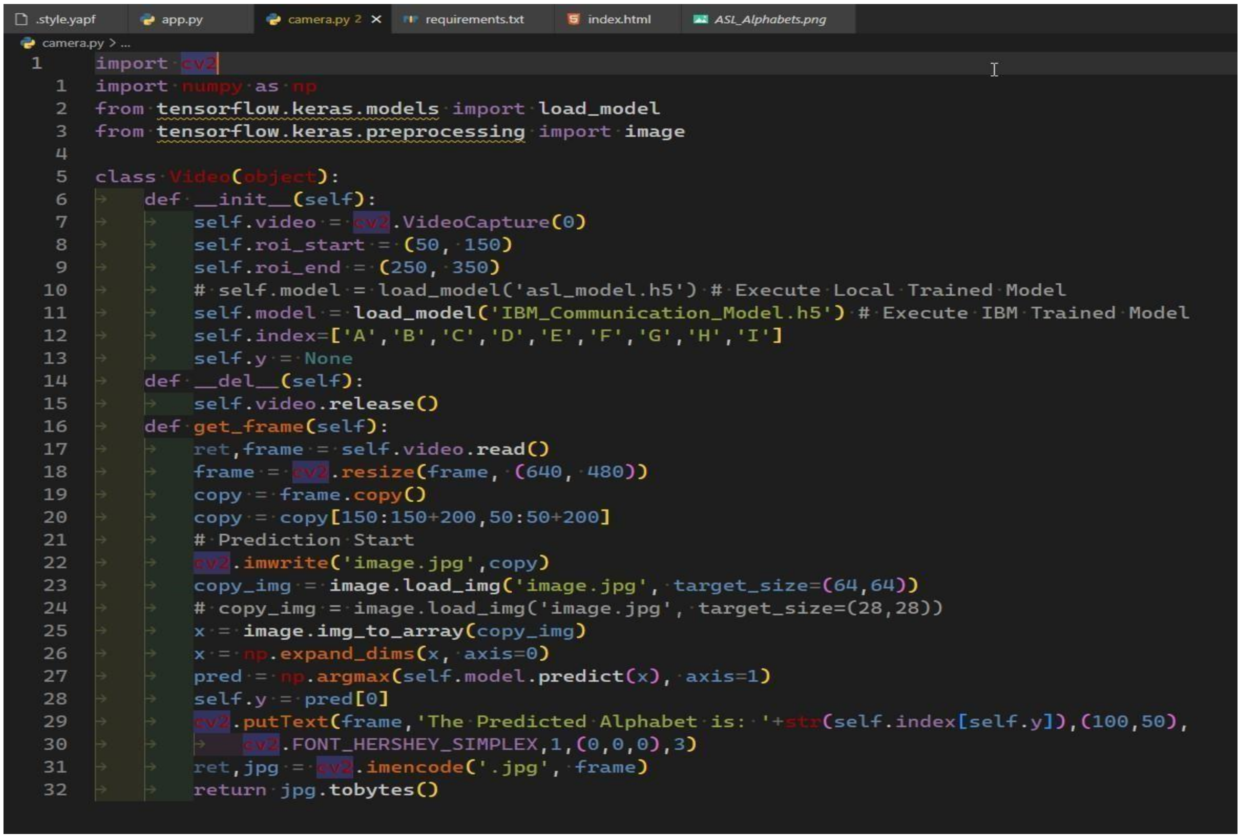
Web app code:

App.py



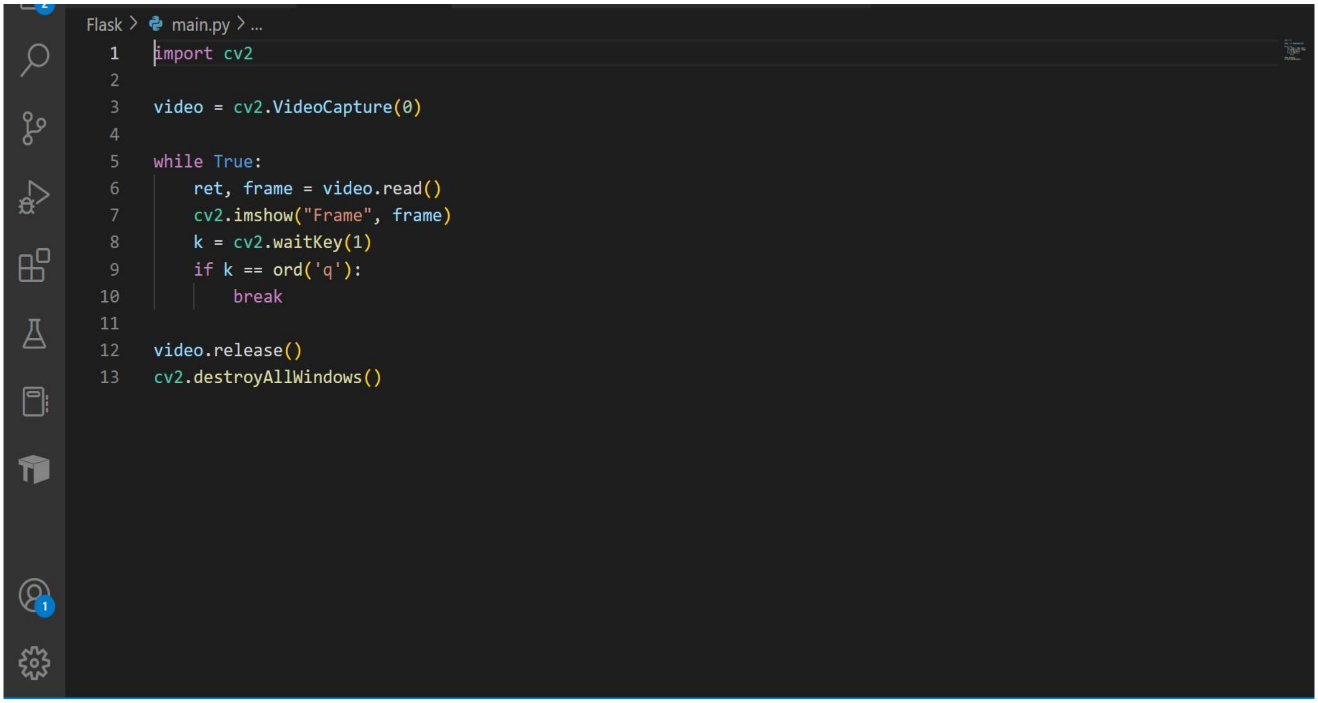
```python
from Flask import Flask, Response, render_template
from flask_socketio import SocketIO, emit
from camera import Video

app = Flask(__name__)
index=['A','B','C','D','E','F','G','H','I']
ll = None
@app.route('/')
def index():
    return render_template('index.html', predict_result=ll)

def gen(camera):
    global ll
    while True:
        frame = camera.get_frame()
        ll = camera.y
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +
            b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    video = Video()
    return Response(gen(video), mimetype='multipart/x-mixed-replace; boundary = frame')
```

Camera.py



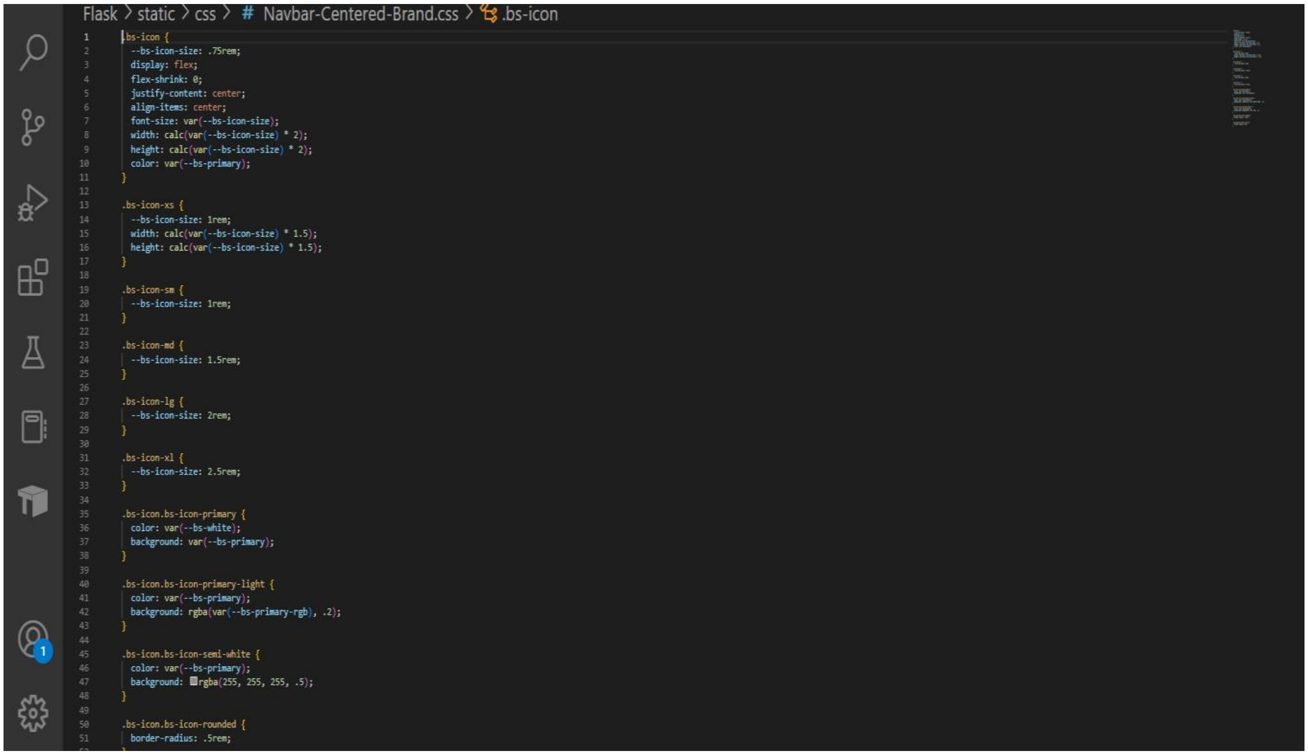
```python
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        # self.model = load_model('asl_model.h5') # Execute Local Trained Model
        self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        # copy_img = image.load_img('image.jpg', target_size=(28,28))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),
            cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

Main.py



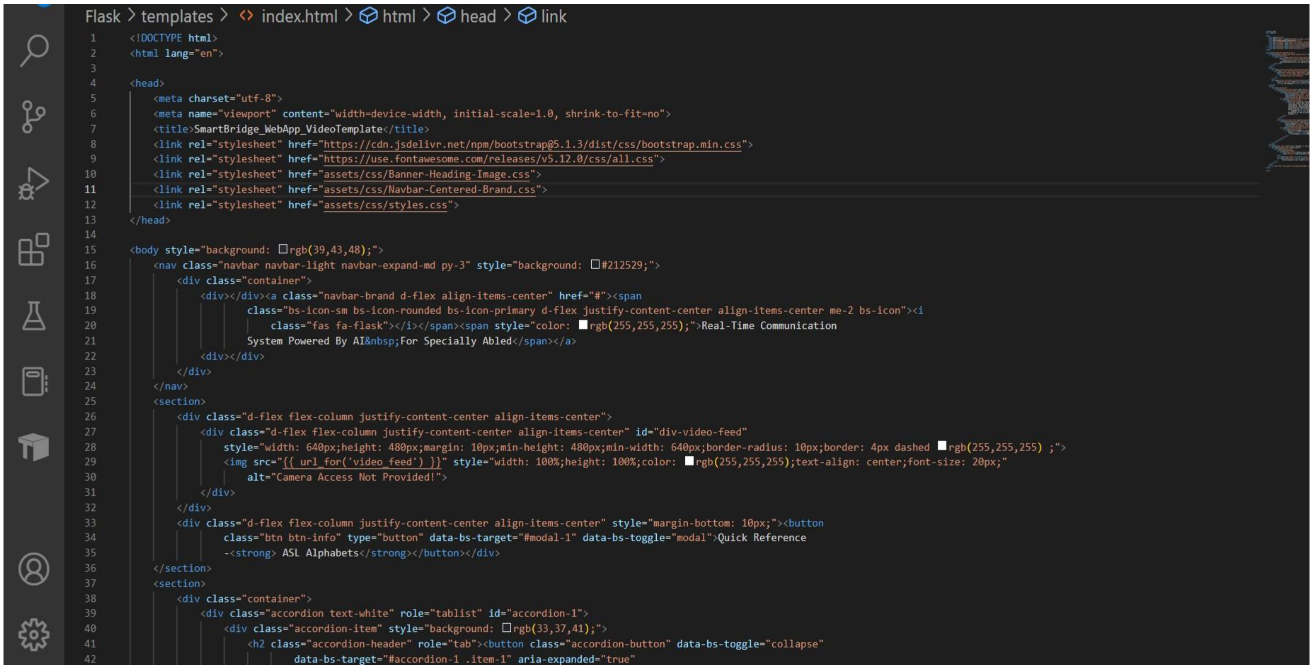
```python
import cv2

video = cv2.VideoCapture(0)

while True:
    ret, frame = video.read()
    cv2.imshow("Frame", frame)
    k = cv2.waitKey(1)
    if k == ord('q'):
        break

video.release()
cv2.destroyAllWindows()
```

Css Code

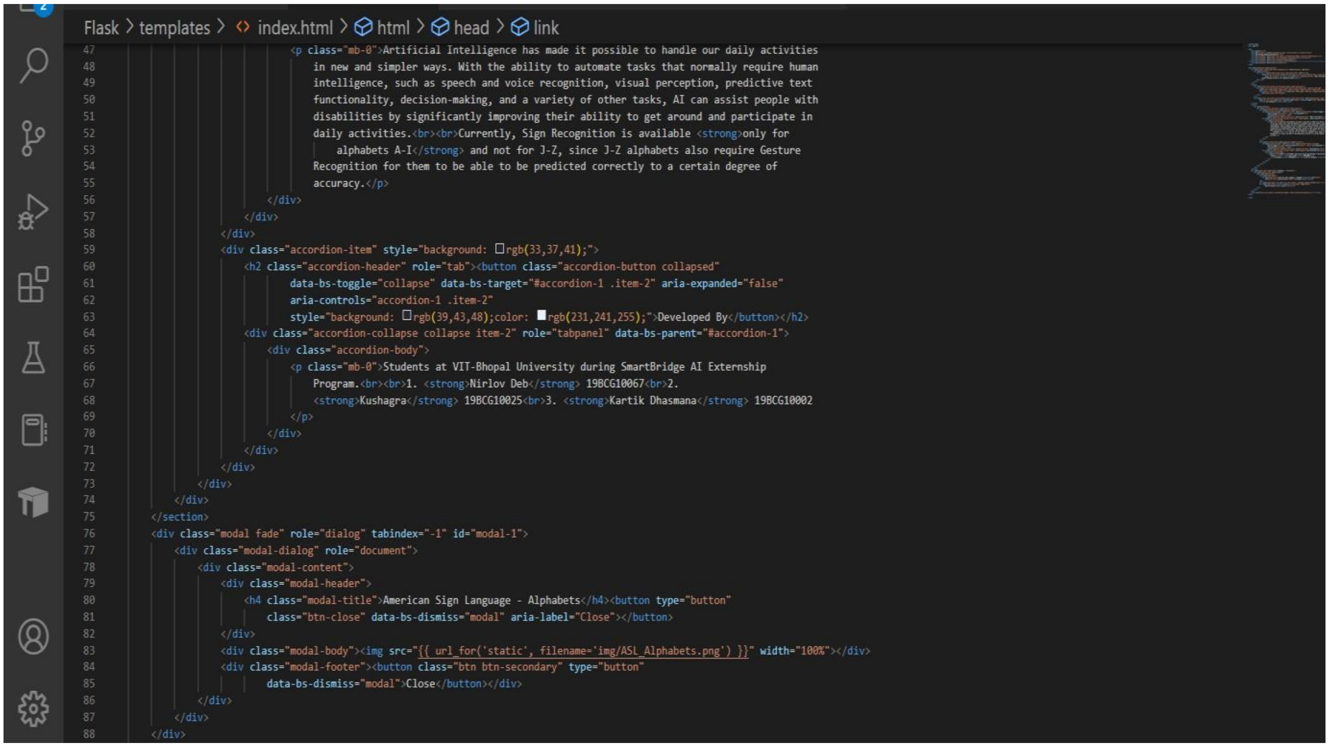

Flask > static > css > # Navbar-Centered-Brand.css > .bs-icon

```css
.bs-icon {
  --bs-icon-size: .75rem;
  display: flex;
  flex-shrink: 0;
  justify-content: center;
  align-items: center;
  font-size: var(--bs-icon-size);
  width: calc(var(--bs-icon-size) * 2);
  height: calc(var(--bs-icon-size) * 2);
  color: var(--bs-primary);
}

.bs-icon-xs {
  --bs-icon-size: 1rem;
  width: calc(var(--bs-icon-size) * 1.5);
  height: calc(var(--bs-icon-size) * 1.5);
}

.bs-icon-sm {
  --bs-icon-size: 1rem;
}

.bs-icon-md {
  --bs-icon-size: 1.5rem;
}

.bs-icon-lg {
  --bs-icon-size: 2rem;
}

.bs-icon-xl {
  --bs-icon-size: 2.5rem;
}

.bs-icon.bs-icon-primary {
  color: var(--bs-white);
  background: var(--bs-primary);
}

.bs-icon.bs-icon-primary-light {
  color: var(--bs-primary);
  background: rgba(var(--bs-primary-rgb), .2);
}

.bs-icon.bs-icon-semi-white {
  color: var(--bs-primary);
  background: ▨rgba(255, 255, 255, .5);
}

.bs-icon.bs-icon-rounded {
  border-radius: .5rem;
```

Index.html

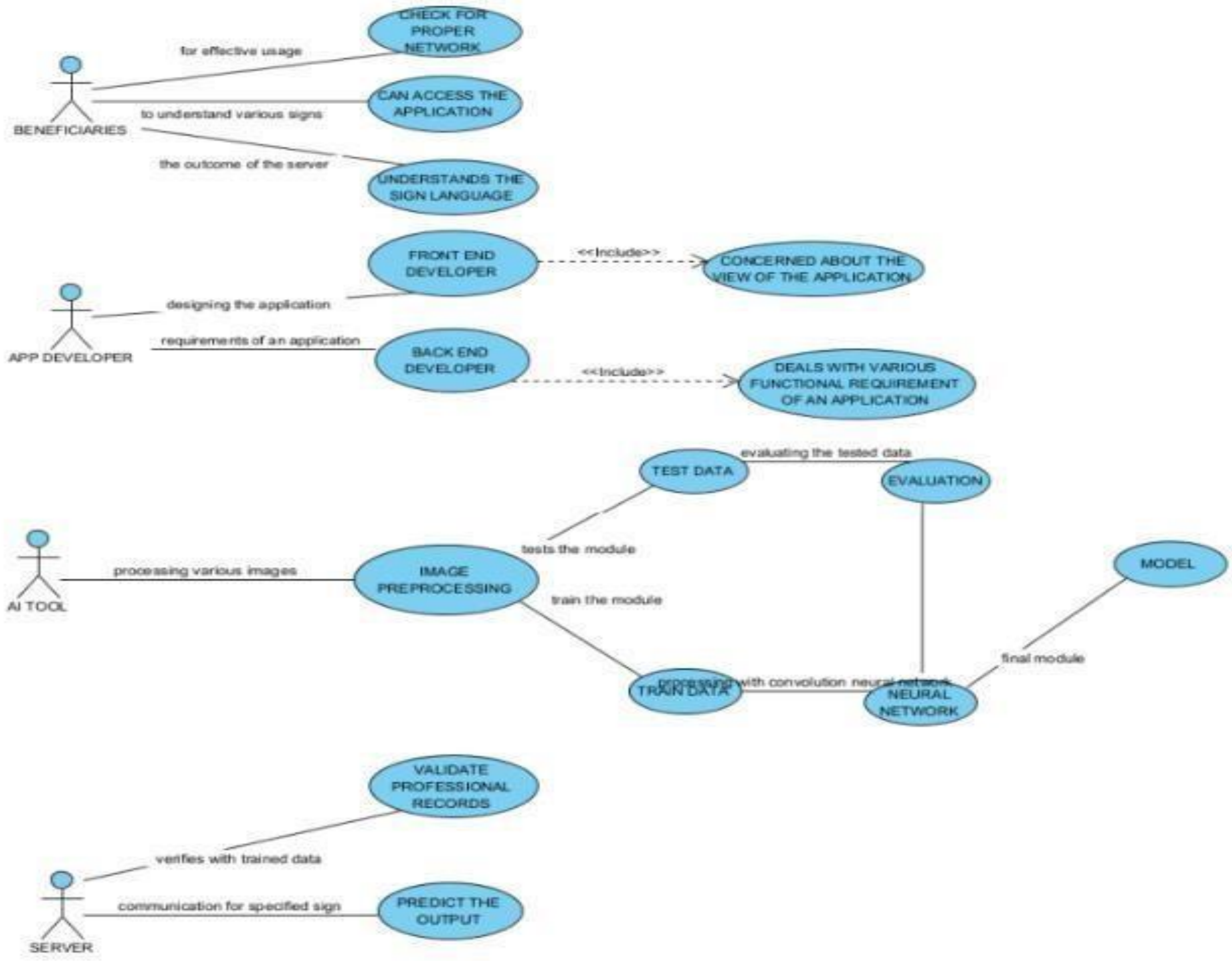


```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>SmartBridge_WebApp_VideoTemplate</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.12.0/css/all.css">
    <link rel="stylesheet" href="assets/css/Banner-Heading-Image.css">
    <link rel="stylesheet" href="assets/css/Navbar-Centered-Brand.css">
    <link rel="stylesheet" href="assets/css/styles.css">
</head>

<body style="background: rgb(39,43,48);">
    <nav class="navbar navbar-light navbar-expand-md py-3" style="background: #212529;">
        <div class="container">
            <div></div><a class="navbar-brand d-flex align-items-center" href="#"><span
                    class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex justify-content-center align-items-center me-2 bs-icon"><i
                        class="fas fa-flask"></i></span><span style="color: rgb(255,255,255);">Real-Time Communication
                    System Powered By AI&nbsp;For Specially Abled</span></a>
            <div></div>
        </div>
    </nav>
    <section>
        <div class="d-flex flex-column justify-content-center align-items-center">
            <div class="d-flex flex-column justify-content-center align-items-center" id="div-video-feed"
                style="width: 640px;height: 480px;margin: 10px;min-height: 480px;min-width: 640px;border-radius: 10px;border: 4px dashed rgb(255,255,255) ;">
                <img src="{{ url_for('video_feed') }}" style="width: 100%;height: 100%;color: rgb(255,255,255);text-align: center;font-size: 20px;"
                    alt="Camera Access Not Provided!">
            </div>
        </div>
        <div class="d-flex flex-column justify-content-center align-items-center" style="margin-bottom: 10px;"><button
                class="btn btn-info" type="button" data-bs-target="#modal-1" data-bs-toggle="modal">Quick Reference
                -<strong> ASL Alphabets</strong></button></div>
    </section>
    <section>
        <div class="container">
            <div class="accordion text-white" role="tablist" id="accordion-1">
                <div class="accordion-item" style="background: rgb(33,37,41);">
                    <h2 class="accordion-header" role="tab"><button class="accordion-button" data-bs-toggle="collapse"
                            data-bs-target="#accordion-1 .item-1" aria-expanded="true"
```



```
                            <p class="mb-0">Artificial Intelligence has made it possible to handle our daily activities
                                in new and simpler ways. With the ability to automate tasks that normally require human
                                intelligence, such as speech and voice recognition, visual perception, predictive text
                                functionality, decision-making, and a variety of other tasks, AI can assist people with
                                disabilities by significantly improving their ability to get around and participate in
                                daily activities.<br><br>Currently, Sign Recognition is available <strong>only for
                                    alphabets A-I</strong> and not for J-Z, since J-Z alphabets also require Gesture
                                Recognition for them to be able to be predicted correctly to a certain degree of
                                accuracy.</p>
                        </div>
                    </div>
                </div>
                <div class="accordion-item" style="background: rgb(33,37,41);">
                    <h2 class="accordion-header" role="tab"><button class="accordion-button collapsed"
                            data-bs-toggle="collapse" data-bs-target="#accordion-1 .item-2" aria-expanded="false"
                            aria-controls="accordion-1 .item-2"
                            style="background: rgb(39,43,48);color: rgb(231,241,255);">Developed By</button></h2>
                    <div class="accordion-collapse collapse item-2" role="tabpanel" data-bs-parent="#accordion-1">
                        <div class="accordion-body">
                            <p class="mb-0">Students at VIT-Bhopal University during SmartBridge AI Externship
                                Program.<br><br>1. <strong>Nirlov Deb</strong> 19BCG10067<br>2.
                                <strong>Kushagra</strong> 19BCG10025<br>3. <strong>Kartik Dhasmana</strong> 19BCG10002
                            </p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>
    <div class="modal fade" role="dialog" tabindex="-1" id="modal-1">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h4 class="modal-title">American Sign Language - Alphabets</h4><button type="button"
                        class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body"><img src="{{ url_for('static', filename='img/ASL_Alphabets.png') }}" width="100%"></div>
                <div class="modal-footer"><button class="btn btn-secondary" type="button"
                        data-bs-dismiss="modal">Close</button></div>
            </div>
        </div>
    </div>
```

7.3 Database Schema


CHECK FOR PROPER NETWORK
for effective usage
CAN ACCESS THE APPLICATION
BENEFICIARIES
to understand various signs
the outcome of the server
UNDERSTANDS THE SIGN LANGUAGE
FRONT END DEVELOPER
<<Include>>
CONCERNED ABOUT THE VIEW OF THE APPLICATION
designing the application
APP DEVELOPER
requirements of an application
BACK END DEVELOPER
<<Include>>
DEALS WITH VARIOUS FUNCTIONAL REQUIREMENT OF AN APPLICATION
evaluating the tested data
TEST DATA
EVALUATION
tests the module
AI TOOL
processing various images
IMAGE PREPROCESSING
MODEL
train the module
final module
TRAIN DATA
processing with convolution neural network
NEURAL NETWORK
VALIDATE PROFESSIONAL RECORDS
verifies with trained data
SERVER
communication for specified sign
PREDICT THE OUTPUT

8. TESTING

8.1.TEST CASES

Test Case ID	Test Scenario	Steps to Execute	Expected Result	Actual Result
1	Verify if user is able to provide camera access.	1. Enter URL and click go. 2. Give Camera access.	Camera is On.	Working as expected.
2	Verify if user is able to get the desirable prediction for the gesture.	1. Enter URL and click go. 2. Give Camera Access. 3. Make Gesture in front of camera.	Alphabet is predicted for the gesture.	Working as expected.

8.2 User Acceptance Testing

- Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	12	1	1	14
External	5	0	0	0	5
Fixed	11	3	2	2	18

Skipped	0	0	2	0	2
Won’t Fix	4	0	0	0	4
Totals	20	15	5	3	43

- Test Case Analysis

Section	**Total Cases**	**Not Tested**	**Fail**	**Pass**
Client Application	5	1	0	5
Security	2	0	0	2
Exception Reporting	2	0	0	2
Final Report Output	9	0	0	9

- The project developed was tested by an end user and the application converts the gestures to its respective alphabet accurately

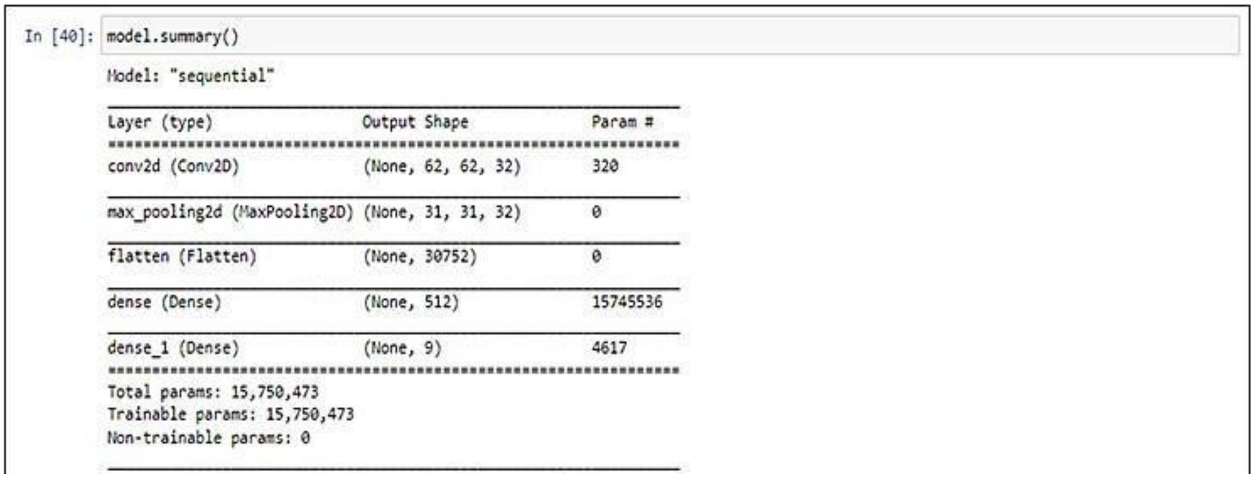
9.RESULTS

The proposed procedure was implemented and tested with set of images. The set of 15750 images of Alphabets from

“A” to “I” are used for training database and a set of 2250 images of Alphabets from “A” to “I” are used for testing database. Once the gesture is recognise the equivalent Alphabet is shown on the screen. Some sample images of the output are provided below:

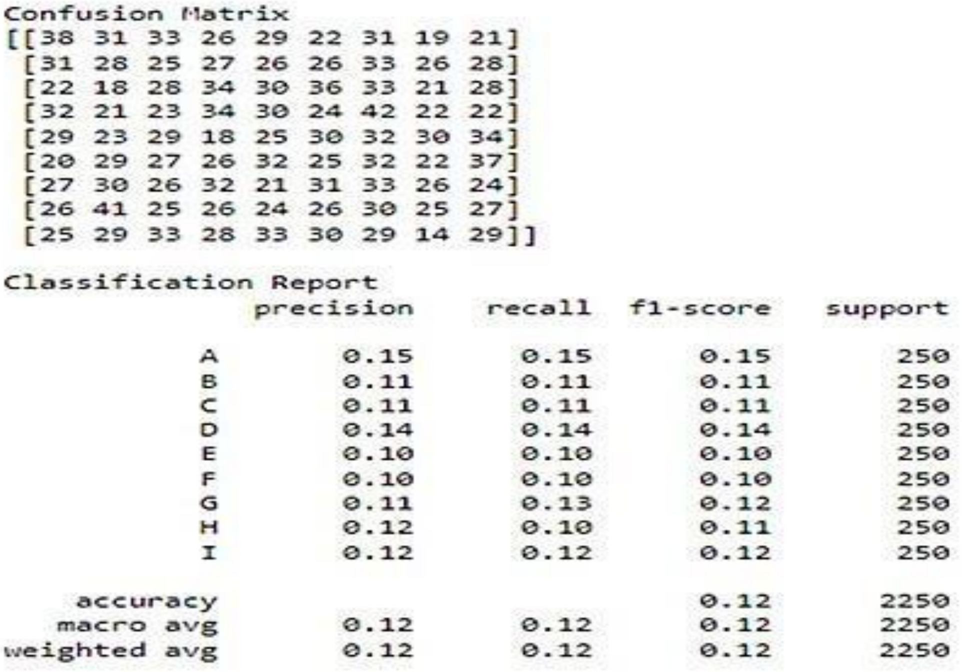
9.1.PERFORMANCE METRICS

- Model Summary



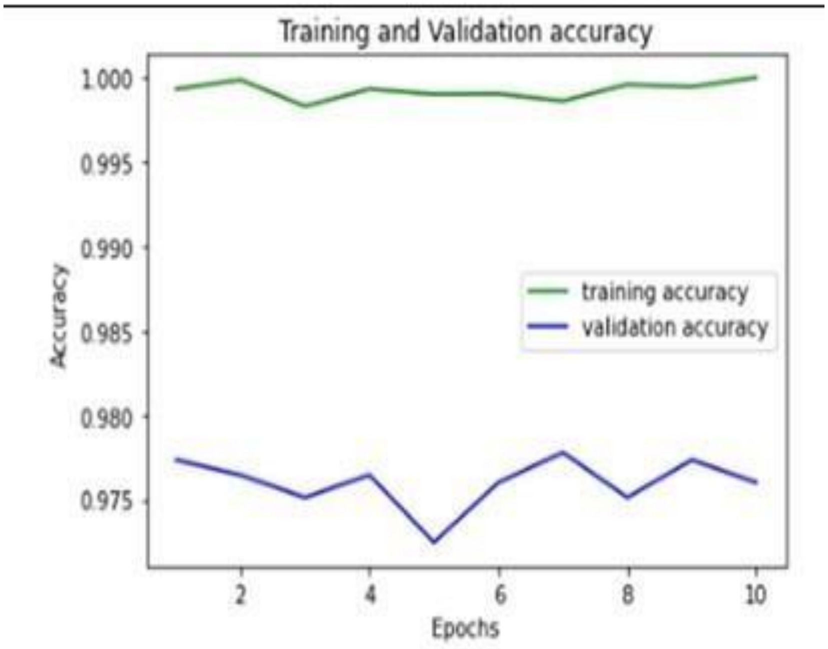
```
In [40]: model.summary()

         Model: "sequential"
         _________________________________________________________________
         Layer (type)                 Output Shape              Param #
         =================================================================
         conv2d (Conv2D)              (None, 62, 62, 32)        320
         _________________________________________________________________
         max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
         _________________________________________________________________
         flatten (Flatten)            (None, 30752)             0
         _________________________________________________________________
         dense (Dense)                (None, 512)               15745536
         _________________________________________________________________
         dense_1 (Dense)              (None, 9)                 4617
         =================================================================
         Total params: 15,750,473
         Trainable params: 15,750,473
         Non-trainable params: 0
         _________________________________________________________________
```

- Confusion Matrix and Classification Report



```
Confusion Matrix
[[38 31 33 26 29 22 31 19 21]
 [31 28 25 27 26 26 33 26 28]
 [22 18 28 34 30 36 33 21 28]
 [32 21 23 34 30 24 42 22 22]
 [29 23 29 18 25 30 32 30 34]
 [20 29 27 26 32 25 32 22 37]
 [27 30 26 32 21 31 33 26 24]
 [26 41 25 26 24 26 30 25 27]
 [25 29 33 28 33 30 29 14 29]]
Classification Report
              precision    recall  f1-score   support

           A       0.15      0.15      0.15       250
           B       0.11      0.11      0.11       250
           C       0.11      0.11      0.11       250
           D       0.14      0.14      0.14       250
           E       0.10      0.10      0.10       250
           F       0.10      0.10      0.10       250
           G       0.11      0.13      0.12       250
           H       0.12      0.10      0.11       250
           I       0.12      0.12      0.12       250

    accuracy                           0.12      2250
   macro avg       0.12      0.12      0.12      2250
weighted avg       0.12      0.12      0.12      2250
```

- Accuracy


Training and Validation accuracy
1.000
0.995
0.990
0.985
0.980
0.975
Accuracy
training accuracy
validation accuracy
2
4
6
8
10
Epochs

10.ADVANTAGES & DISADVANTAGES

Advantages:

1. It is possible to create a mobile application to bridge the communication gap between deaf and dumb persons and the general public.
2. As different sign language standards exist, their dataset can be added, and the user can choose which sign language to read.

Disadvantages:

1. The current model only works from alphabets A to I.
2. In absence of gesture recognition, alphabets from J cannot be identified as they require some kind of gesture input from the user.
3. As the quantity/quality of images in the dataset is low, the accuracy is not great, but that can easily be improved by change in dataset.

11.CONCLUSION

Sign language is a useful tool for facilitating communication between deaf and hearing people. Because it allows for two-way communication, the system aims to bridge the communication gap between deaf people and the rest of society. The proposed methodology translates language into English alphabets that are understandable to humans.

This system sends hand gestures to the model, who recognises them and displays the equivalent Alphabet on the screen. Deaf-mute people can use their hands to perform sign language, which will then be converted into alphabets, thanks to this project.

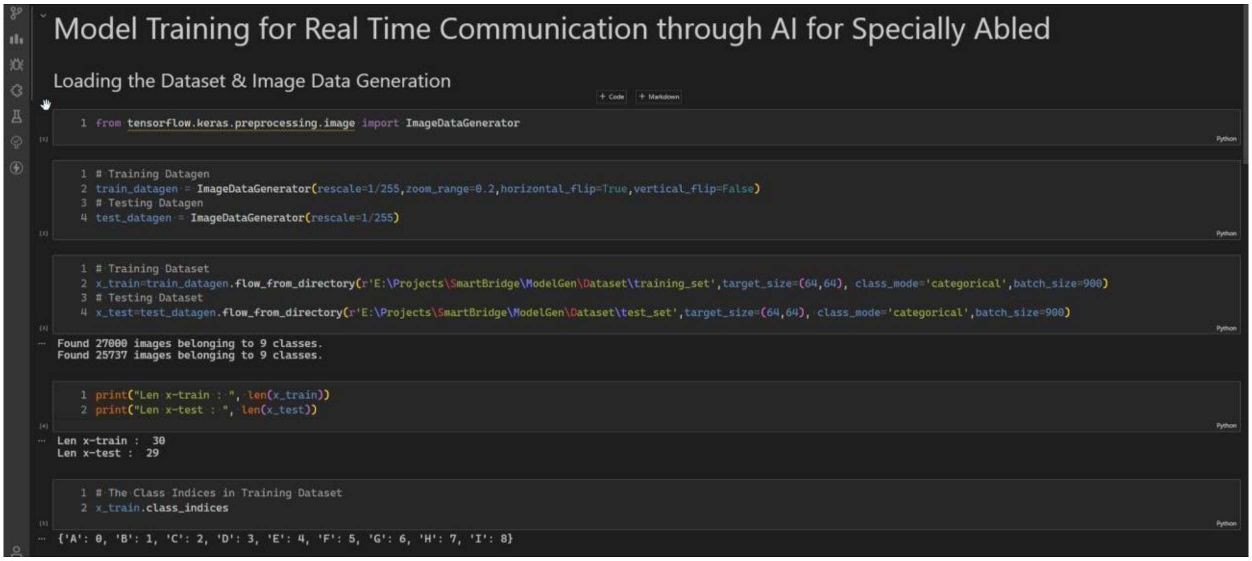
12.FUTURE SCOPE

Having a technology that can translate hand sign language to its corresponding alphabet is a game changer in the field of communication and Ai for the specially abled people such as deaf and dumb. With introduction of gesture recognition, the web app can easily be expanded to recognize letters beyond 'I', digits and other symbols plus gesture recognition can also allow controlling of software/hardware interface

13.APPENDIX

Source Code

Model Training and Saving



Model Training for Real Time Communication through AI for Specially Abled

Loading the Dataset & Image Data Generation

```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```python
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```python
# Training Dataset
x_train=train_datagen.flow_from_directory(r'E:\Projects\SmartBridge\ModelGen\Dataset\training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

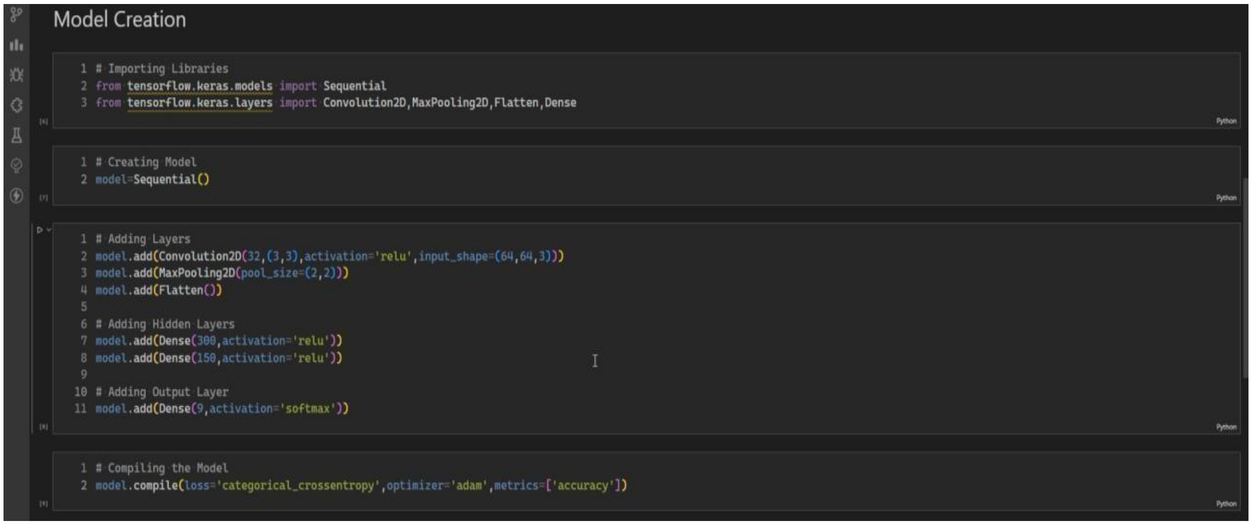
```
Found 27000 images belonging to 9 classes.
Found 25737 images belonging to 9 classes.
```

```python
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train :  30
Len x-test :  29
```

```python
# The Class Indices in Training Dataset
x_train.class_indices
```

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```



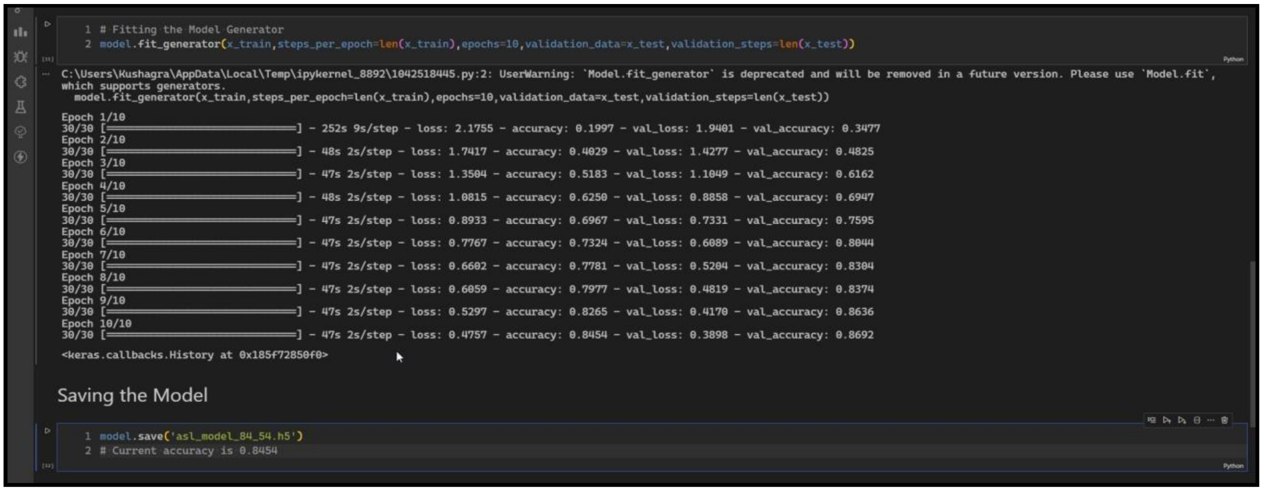
Model Creation

```python
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

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

```python
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

# Adding Hidden Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))

# Adding Output Layer
model.add(Dense(9,activation='softmax'))
```

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



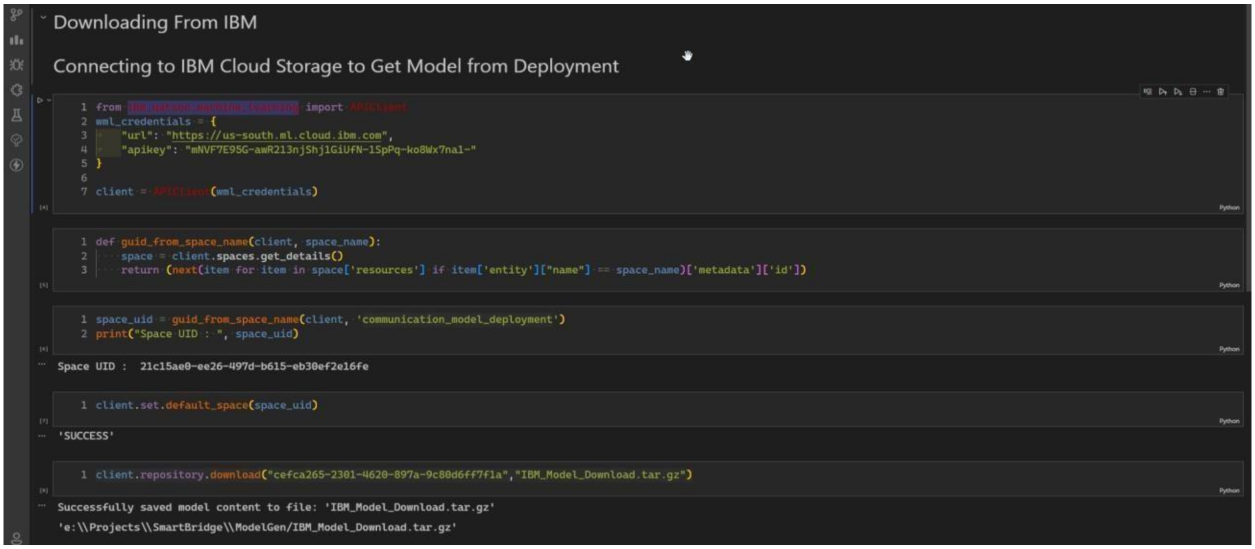
```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

```
C:\Users\Kushagra\AppData\Local\Temp\ipykernel_8892\1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
Epoch 1/10
30/30 [==============================] - 252s 9s/step - loss: 2.1755 - accuracy: 0.1997 - val_loss: 1.9401 - val_accuracy: 0.3477
Epoch 2/10
30/30 [==============================] - 48s 2s/step - loss: 1.7417 - accuracy: 0.4029 - val_loss: 1.4277 - val_accuracy: 0.4825
Epoch 3/10
30/30 [==============================] - 47s 2s/step - loss: 1.3504 - accuracy: 0.5183 - val_loss: 1.1049 - val_accuracy: 0.6162
Epoch 4/10
30/30 [==============================] - 48s 2s/step - loss: 1.0815 - accuracy: 0.6250 - val_loss: 0.8858 - val_accuracy: 0.6947
Epoch 5/10
30/30 [==============================] - 47s 2s/step - loss: 0.8933 - accuracy: 0.6967 - val_loss: 0.7331 - val_accuracy: 0.7595
Epoch 6/10
30/30 [==============================] - 47s 2s/step - loss: 0.7767 - accuracy: 0.7324 - val_loss: 0.6089 - val_accuracy: 0.8044
Epoch 7/10
30/30 [==============================] - 47s 2s/step - loss: 0.6602 - accuracy: 0.7781 - val_loss: 0.5204 - val_accuracy: 0.8304
Epoch 8/10
30/30 [==============================] - 47s 2s/step - loss: 0.6059 - accuracy: 0.7977 - val_loss: 0.4819 - val_accuracy: 0.8374
Epoch 9/10
30/30 [==============================] - 47s 2s/step - loss: 0.5297 - accuracy: 0.8265 - val_loss: 0.4170 - val_accuracy: 0.8636
Epoch 10/10
30/30 [==============================] - 47s 2s/step - loss: 0.4757 - accuracy: 0.8454 - val_loss: 0.3898 - val_accuracy: 0.8692

<keras.callbacks.History at 0x185f72850f0>
```

Saving the Model

```
model.save('asl_model_84_54.h5')
# Current accuracy is 0.8454
```

__IBM Model Training & Download Code:__



Downloading From IBM

Connecting to IBM Cloud Storage to Get Model from Deployment

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "mNVF7E95G-awR213njShj1GiUfN-1SpPq-ko8Wx7nal-"
}

client = APIClient(wml_credentials)
```

```
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return (next(item for item in space['resources'] if item['entity']["name"] == space_name)['metadata']['id'])
```

```
space_uid = guid_from_space_name(client, 'communication_model_deployment')
print("Space UID : ", space_uid)
```

```
Space UID :  21c15ae0-ee26-497d-b615-eb30ef2e16fe
```

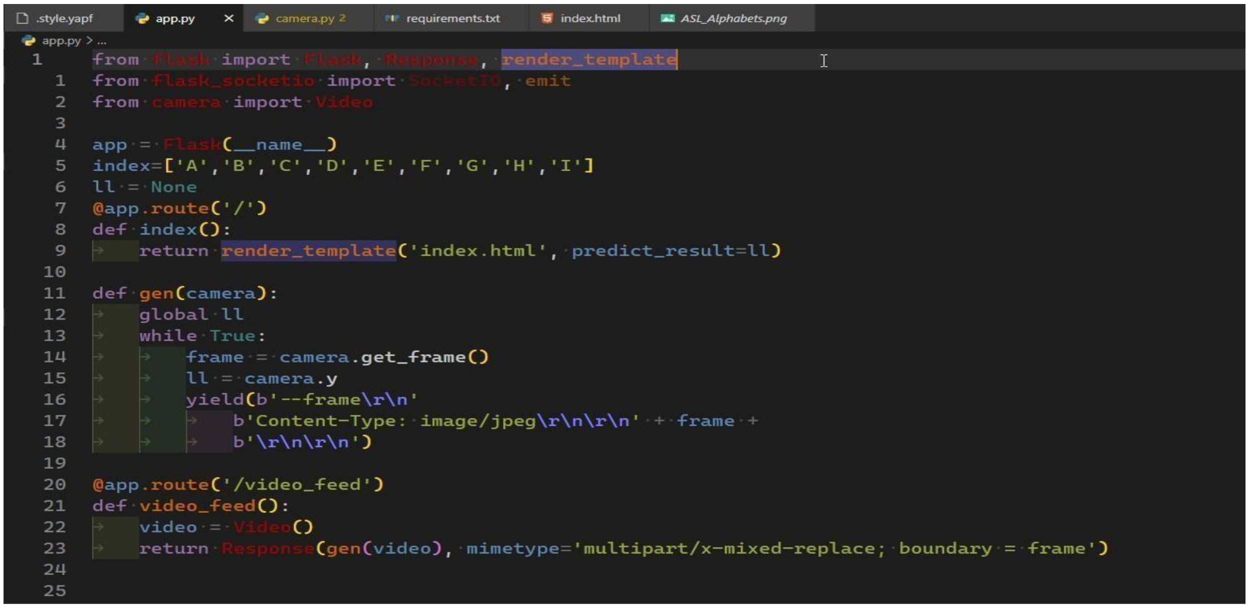
```
client.set.default_space(space_uid)
```

```
'SUCCESS'
```

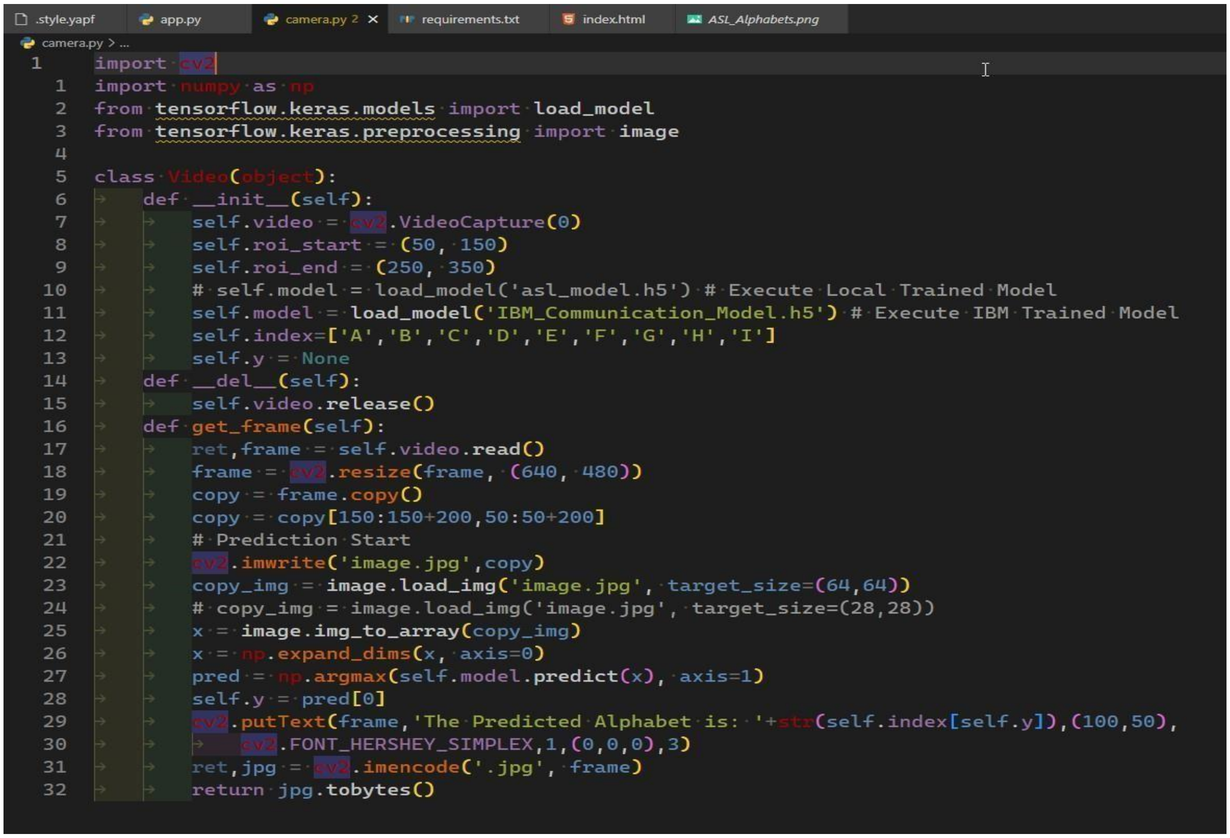
```
client.repository.download("cefca265-2301-4620-897a-9c80d6ff7f1a","IBM_Model_Download.tar.gz")
```

```
Successfully saved model content to file: 'IBM_Model_Download.tar.gz'
'e:\\Projects\\SmartBridge\\ModelGen/IBM_Model_Download.tar.gz'
```

__Web app code:__

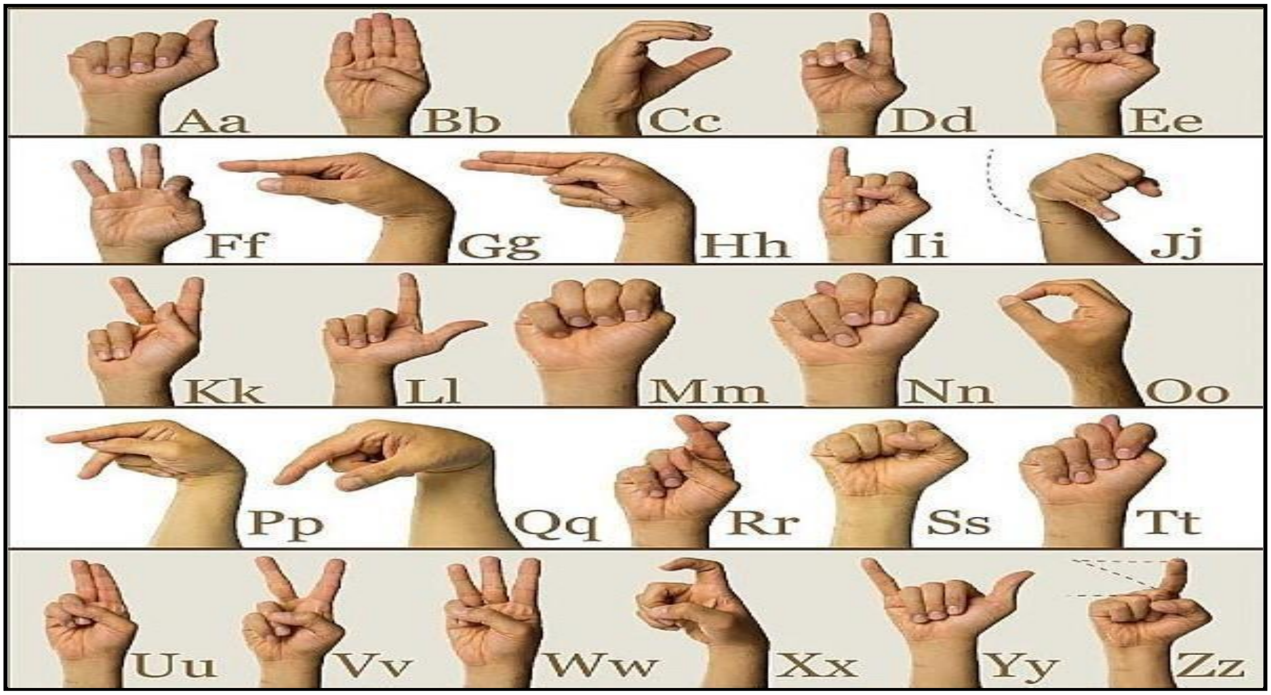


```
from flask import Flask, Response, render_template
from flask_socketio import SocketIO, emit
from camera import Video

app = Flask(__name__)
index=['A','B','C','D','E','F','G','H','I']
ll = None
@app.route('/')
def index():
    return render_template('index.html', predict_result=ll)

def gen(camera):
    global ll
    while True:
        frame = camera.get_frame()
        ll = camera.y
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +
            b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    video = Video()
    return Response(gen(video), mimetype='multipart/x-mixed-replace; boundary = frame')


```



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

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        # self.model = load_model('asl_model.h5') # Execute Local Trained Model
        self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        # copy_img = image.load_img('image.jpg', target_size=(28,28))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),
            cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

American Sign Language Standard Reference:



Github Link

- **https://github.com/IBM-EPBL/IBM-Project-1587-1658399807**

Project Demo Link

- https://youtu.be/MeQhFdUOJGU