REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

IBM PROJECT REPORT

Submitted by

POOJA.S (TL)	**(312819205028)**
AKILANDESWARI.A	**(312819205005)**
KAVIYADHARSHINI.S	**(312819205012)**
YUVASRI.S	**(312819205050)**
PRINCY.F	**(312819205033)**

BACHELOR OF TECHNOLOGY
IN

INFORMATION TECHNOLOGY

AGNI COLLEGE OF TECHNOLOGY ,CHENNAI

ANNA UNIVERSITY-CHENNAI 600025

OCT 2022

Table Of Contents

1. INTRODUCTION

1.1 Project Overview

People get to know one another by sharing their ideas, thoughts, and experiences with those around them. There are numerous ways to accomplish this, the best of which is the gift of "Speech." Everyone can very convincingly transfer their thoughts and understand each other through speech. It will be unjust if we overlook those who are denied this priceless gift: the deaf and dumb. In such cases, the human hand has remained the preferred method of communication.

1.2 Purpose

The project's purpose is to create a system that translates sign language into a human understandable language so that ordinary people may understand it.

2. LITERATURE SURVEY

2.1 Existing problem

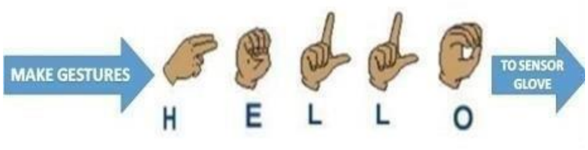
Some of the existing solutions for solving this problem are:

Technology

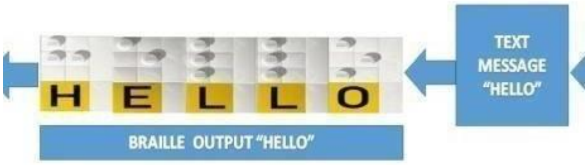
One of the easiest ways to communicate is through technology such as a smart phone or laptop. A deaf person can type out what they want to say and a person who is blind or has low vision can use a screen reader to read the text out loud. A blind person can also use voice recognition software to convert 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 both people and their environment (for example, some places may have too much background noise).



ɔN



B

2.2 REFERENCES

[1] Prof. P.G. Ahire, K.B. Tilekary,T.A. Jawake, P.B. Warale, "Two Way Communicator between Deaf and Dumb People and Normal People", 978-1-4799-6892-3/15 31.00 c 2015 IEEE.

[2] Shreyashi Narayan Sawant, "Sign Language recognition System to aid Deaf- dumb People Using PCA", IJCSET ISSN : 2229-3345 Vol. 5 No. 05 May 2014.

[3] Amitkumar Shinde, Ramesh Kagalkar, "Sign Language to Text and Vice Versa Recognition using Computer Vision in Marathi", International Journal of Computer Applications (0975 – 8887) National Conference on Advances in Computing (NCAC 2015)

[4] Setiawardhana, Rizky Yuniar Hakkun, Achmad Baharuddin, "Sign Language Learning based on Android For

Deaf and Speech Impaired People", 978-1-4673-9345-

4/15/31.00 c 2015 IEEE

[5] M. Ebrahim Al-Ahdal & Nooritawati Md Tahir,'' Review in

Sign Language Recognition Systems'' Symposium on

Computer & Informatics (ISCI),pp:52-57, IEEE ,2012

[6] Archana S. Ghotkar, Rucha Khatal, Sanjana Khupase,

Surbhi Asati & Mithila Hadap,'' Hand Gesture Recognition for

Indian Sign Language'' International Conference on Computer Communication and Informatics (ICCCI), pp:14.IEEE,Jan 2012.

[7] Iwan Njoto Sandjaja, Nelson Marcos,'' Sign Language Number Recognition'' Fifth International Joint Conference on

INC, IMS and IDC, IEEE 2009

2.3 PROBLEM STATEMENT DEFINITION

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.

Depending on the type of disability and profile, communicating with others can be a challenge. The same holds true for staying connected to others in a world that's more and more digitized with the growing importance of social media and our dependence to the Internet. But technology and AI leave no one behind and can be at the service of people with disabilities. A lot of apps use artificial intelligence to favor accessibility

AI has a huge impact on people with disabilities' everyday lives: a person with a mental impairment can easily comprehend the world around him thanks to text summarization. What may at first be a complicated message to decipher turns out to be an easy-tounderstand text. Things that at first were difficult or impossible for them are now easily accessible on a daily basis. AI enables people with disabilities to step into a world where their difficulties are understood and taken into account. Technology adapts and helps transform the world into an inclusive place with artificial intelligence accessibility. There is a certain sense of equality as AI puts everybody, with or without disabilities, at the same level.

We've seen the main points regarding AI accessibility but concretely, where is AI put into action to improve people with disabilities' lives? How does AI help them remain autonomous? Let's focus on 4 major situations where AI adds value

3. IDEATION & PROPOSED SOLUTION

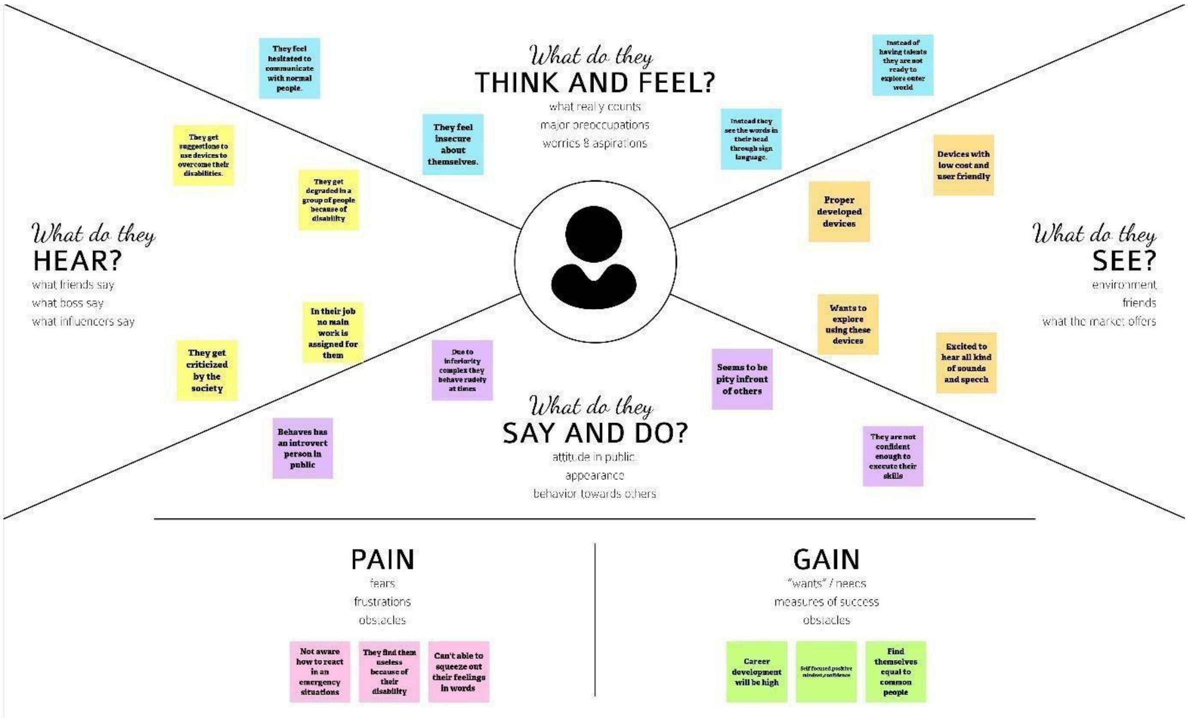
3.1 EMPATHY MAP CANVAS

Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.



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

They feel hesitated to communicate with normal people.

They feel insecure about themselves.

Instead they see the words in their head through sign language.

Instead of having talents they are not ready to explore outer world

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

They get suggestions to use devices to overcome their disabilities.

They get degraded in a group of people because of disability

In their job no main work is assigned for them

They get criticized by the society

What do they SEE?
environment
friends
what the market offers

Devices with low cost and user friendly

Proper developed devices

Wants to explore using these devices

Excited to hear all kind of sounds and speech

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

Due to inferiority complex they behave rudely at times

Behaves has an introvert person in public

Seems to be pity infront of others

They are not confident enough to execute their skills

PAIN
fears
frustrations
obstacles

Not aware how to react in an emergency situations

They find them useless because of their disability

Can't able to squeeze out their feelings in words

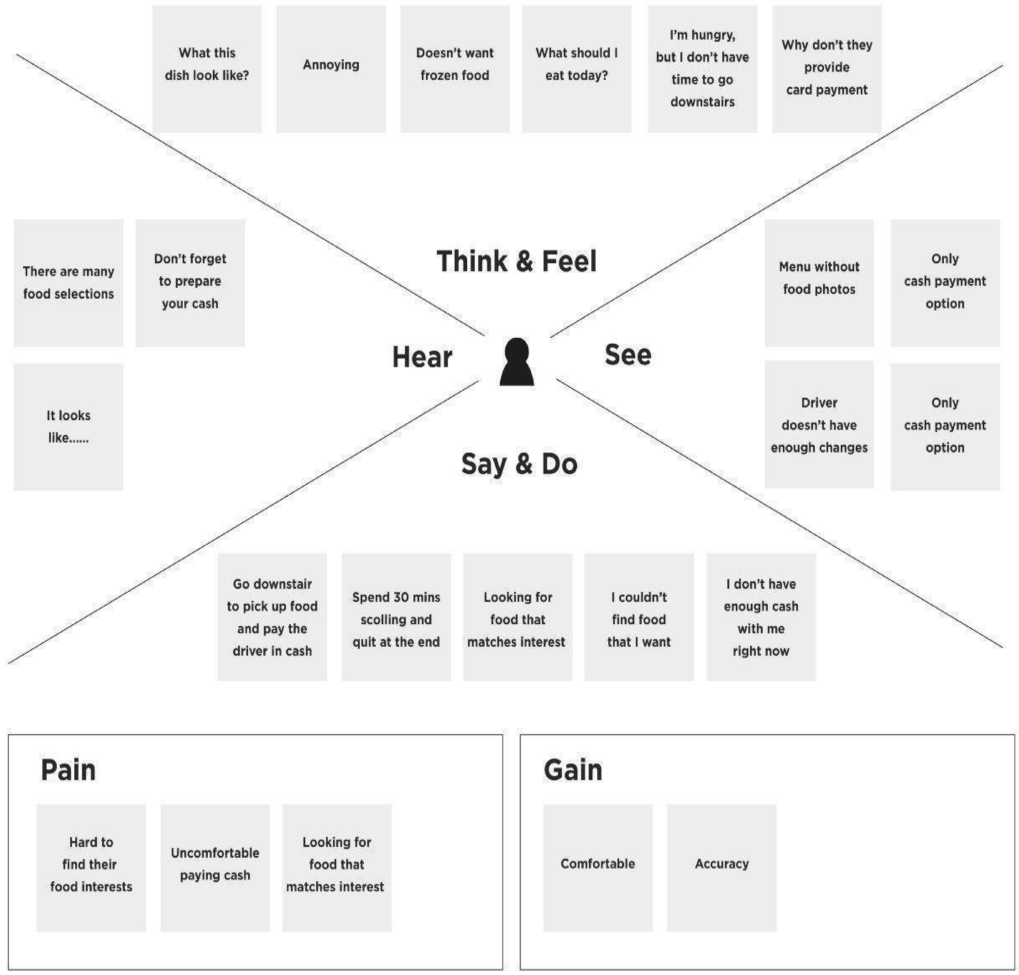
GAIN
"wants" / needs
measures of success
obstacles

Career development will be high

Self focused,positive mindset,confidence

Find themselves equal to common people

Example: Food Ordering & Delivery Application



Think & Feel

What this dish look like?

Annoying

Doesn't want frozen food

What should I eat today?

I'm hungry, but I don't have time to go downstairs

Why don't they provide card payment

Hear

There are many food selections

Don't forget to prepare your cash

It looks like……

See

Menu without food photos

Only cash payment option

Driver doesn't have enough changes

Only cash payment option

Say & Do

Go downstair to pick up food and pay the driver in cash

Spend 30 mins scolling and quit at the end

Looking for food that matches interest

I couldn't find food that I want

I don't have enough cash with me right now

Pain

Hard to find their food interests

Uncomfortable paying cash

Looking for food that matches interest

Gain

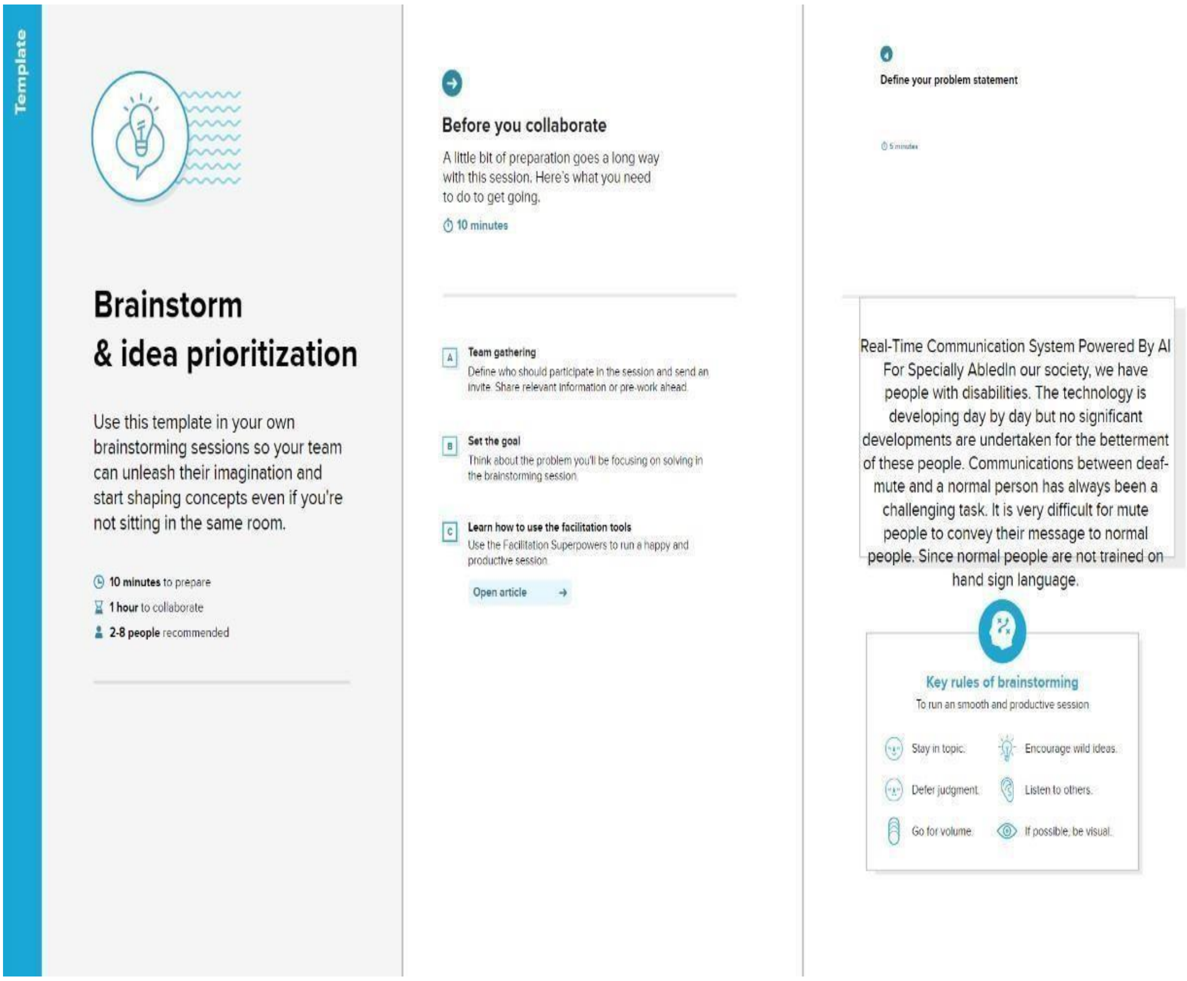
Comfortable

Accuracy

3.2 IDEATION & BRAINSTRORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



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

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.

Define your problem statement

5 minutes

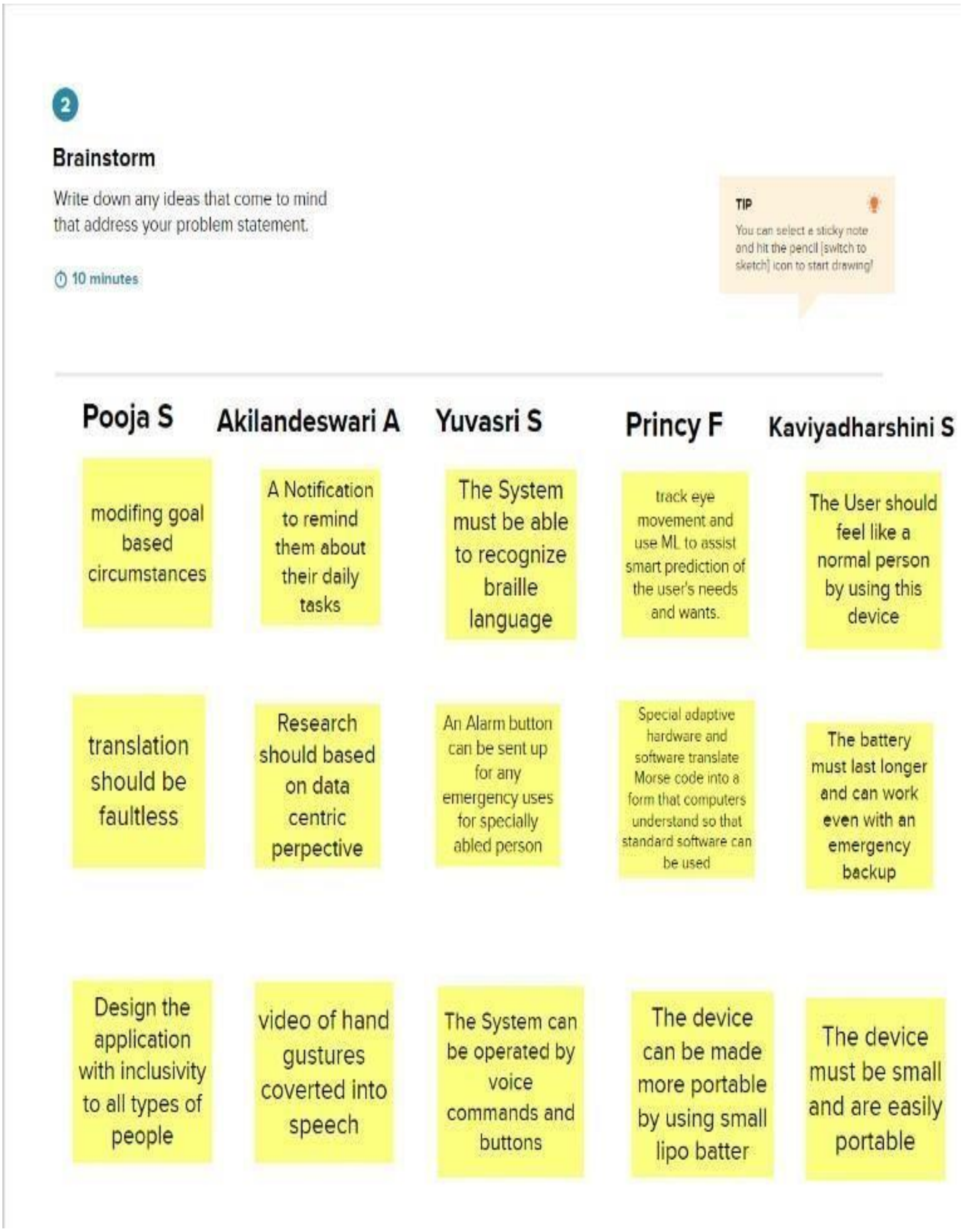
Real-Time Communication System Powered By AI For Specially AbledIn our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always 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.

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.

Step-2: Brainstorm, Idea Listing and Grouping



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!

Pooja S	Akilandeswari A	Yuvasri S	Princy F	Kaviyadharshini S
modifing goal based circumstances	A Notification to remind them about their daily tasks	The System must be able to recognize braille language	track eye movement and use ML to assist smart prediction of the user's needs and wants.	The User should feel like a normal person by using this device
translation should be faultless	Research should based on data centric perpective	An Alarm button can be sent up for any emergency uses for specially abled person	Special adaptive hardware and software translate Morse code into a form that computers understand so that standard software can be used	The battery must last longer and can work even with an emergency backup
Design the application with inclusivity to all types of people	video of hand gustures coverted into speech	The System can be operated by voice commands and buttons	The device can be made more portable by using small lipo batter	The device must be small and are easily portable

3.3 PROPOSED SOLUTION

PROPOSED SOLUTION

The portable device is the transduction system and consists of a glove comprising 5 flex sensors connected to a microcontroller through front-end electronics. The sensors are positioned in the glove hence the finger joints are in the middle of the sensitive area this way the sensor covers the finger joint area and follows its movement. Doing sign language our finger will bend, each bending of finger the sensor send signal to our controller. Our controller match the sensor signals with train signals. Recorded audio sound will playback for each sign. Opposite person easily communicate to deaf a keypad and display is used. Hearing impaired users can use text descriptions and icons. This module makes the device handy for the vocally disabled as it enables them to vocalise words by typing it on the screen.

NOVELTY

There are few mobile applications for Deaf and dumb like Deaf and Dumb through 4G applications. These techniques only enable communication between deaf and dumb through sign language using mobile phones. The mobile application which proposed in helps to make recognition of sign language. Mobile-based Deaf and Dumb Interaction System project in proposed mobile application that enables the needs of deaf and dumb developing a voice-activated mobile which would convert their sign language into messages that may be read by other users, this message can also converted to a voice.

FEASIBILITY OF IDEA

Without dialling number we can communicate to other like face to face communication. It does not require large amount of storage as it uses the Hand speak support through online. The sign words are signed in the same order as letters appear in English alphabets. This project prepares individuals to work as interpreter/translators facilitating and mediating communication between Deaf/Dumb and hearing people.

BUSINESS MODEL

The technology used are, Motion gesture, Touchless, Infrared Array, Ultrasonic Technology, 2D camera based technology, 3D vision technology. Lip- reading recognition for people with a hearing impairment, Real-time captioning or translations for people with a hearing impairment or even people who don't speak the language. The system contains Real time functioning, Portable, Doesn't damage through use. It provides all the data needed more accurately as it also provides fingers movement data. Most deaf people use a combination of sign language, lip-reading and written communication to go about their daily lives. By using camera is that it removes that needs of sensors gloves and reduce cost from building the system.

SOCIAL IMPACT

Deaf and dumb gesture recognition system is an inexpensive device. It is also very efficient device when compared to the existing devices**.** The main objective of this project is to achieve communication of deaf-mute people like a normal person. It allows deaf and dumb people to communicate with others. It is a best device for these people to overcome their disability. They can express their views to others

SCALABILITY OF SOLUTION

GTTS It is a Python library and CLI tool to interface with Google Translates text- to speech API. Writes spoken mp3 data to a file, a file-like object (byte string) for further audio manipulation, or studio. Implementation on Android platform for smart phones and tablets PCs. Read and write images and detection of images for extracting its features. Detection of shapes like Circle, rectangle etc. in an image, Detection of coin in images. Image processing works with text recognition in images. e.g. Reading Number Plates, Modifying image quality and colour. This describes a project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

3.4 PROBLEM SOLUTION FIT

Hand gesture recognition system is widely used technology for helping the deaf and dumb people. Human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. We aim for developing an deaf and dumb gesture recognize system for establishing communication between the deaf and the dumb people. Gestures are considered as the most natural expressive way for communications between human and computers in virtual system. Hand gestures which can represent ideas using unique shapes and finger orientation have a scope for human machine interaction.

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Sub-Task)
FR-1	User Registration	Registration is done through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Communication requirement	For one on one mentoring, teacher will be available.
FR-4	User requirement	Option should be shown for hand sign to text and voice conversion and vice versa.
FR-5	User Communication	Communication can be done through pc or mobile.
FR-6	Regulatory requirements	In case of any cyber attacks the app gets automatically shut down.
FR-7	Reporting	Automated notification will be received by the developer in case of any issues.
FR-8	Compliance to rules or law	Terms and conditions, private policy, End user subscription agreement and cookies.

4.2 NON FUNCTIONAL REQUIREMENT

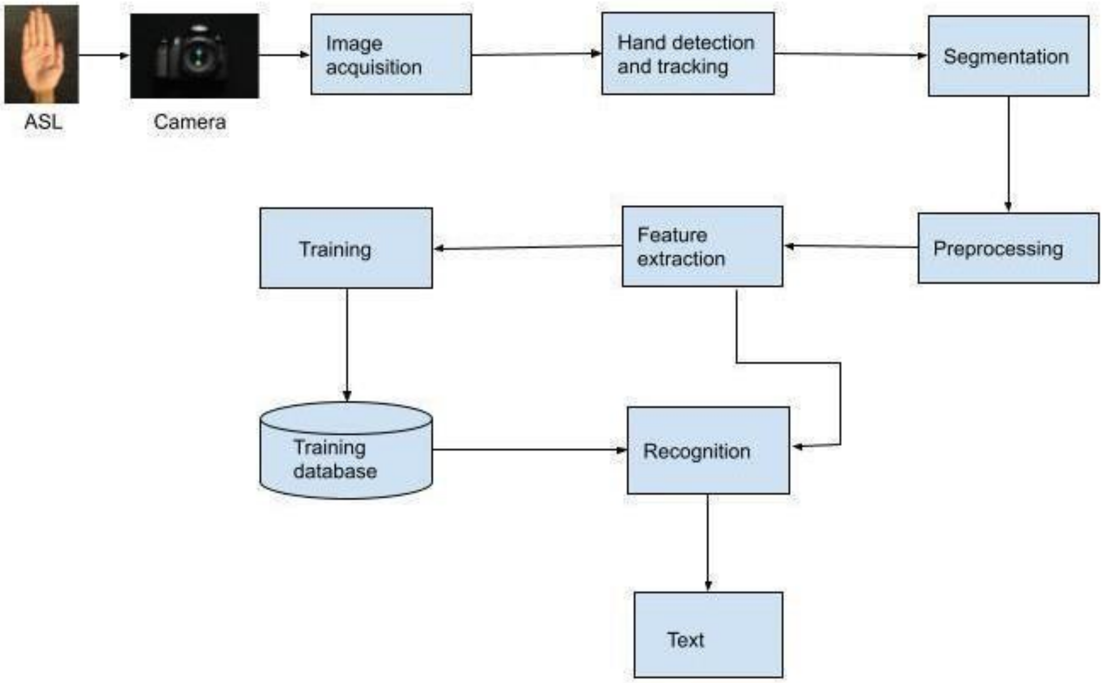
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The camera captures all expressions including facial expressions and hand gestures which can be easily used by all age groups. It can be used by deaf-mute people and their care takers.
NFR-2	**Security & Privacy**	The system is more secure and information of the customers is also maintained confidentially.
NFR-3	**Accuracy**	The system must have a great accuracy rate. The accuracy is important so that the disabled students could get a clear understanding.
NFR-4	**Performance**	The performance of the model is efficient. The cost-effective nature of the system makes it extremely liable. The latency is very less for the conversion process.
NFR-5	**Availability**	The solution is suitable for different languages and can be used in many countries. It can be trained for all the available sign languages. This model can be used at any time anywhere.

NFR-6	**Scalability**	The system gives output rapidly. It also predicts quickly when it gets so many inputs at a time. It predicts different types of sign language at a time. Upto 25000 users can be use this model at a time.

5. PROJECT DESIGN

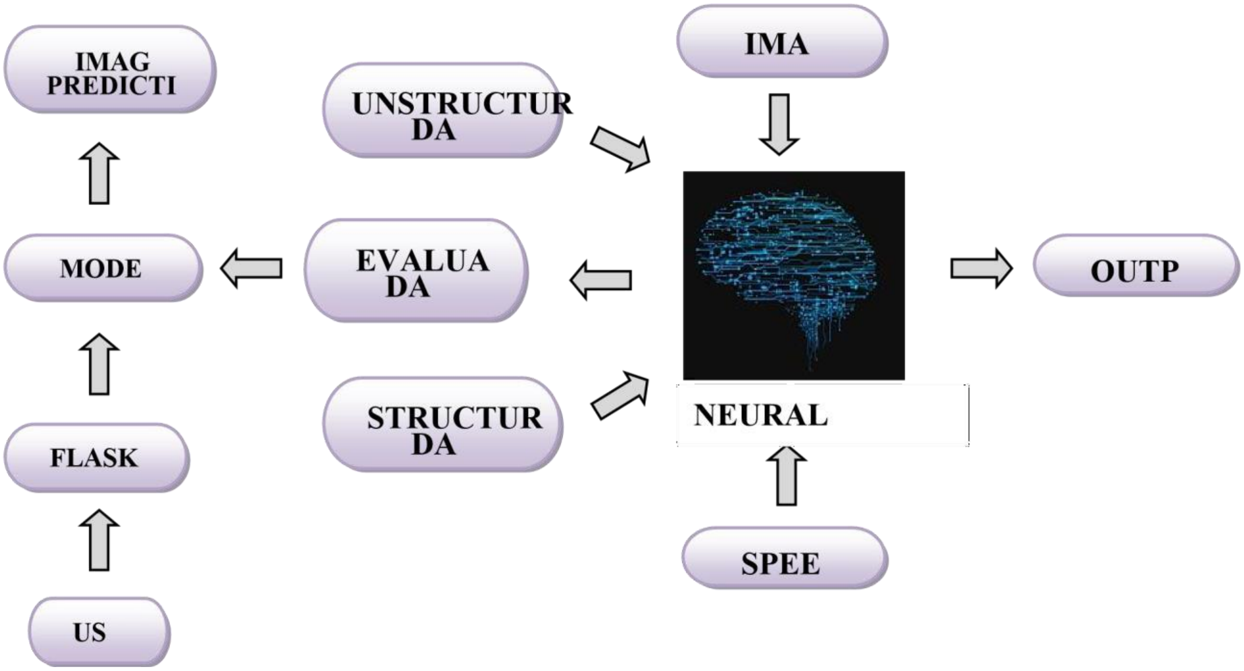
5.1 DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

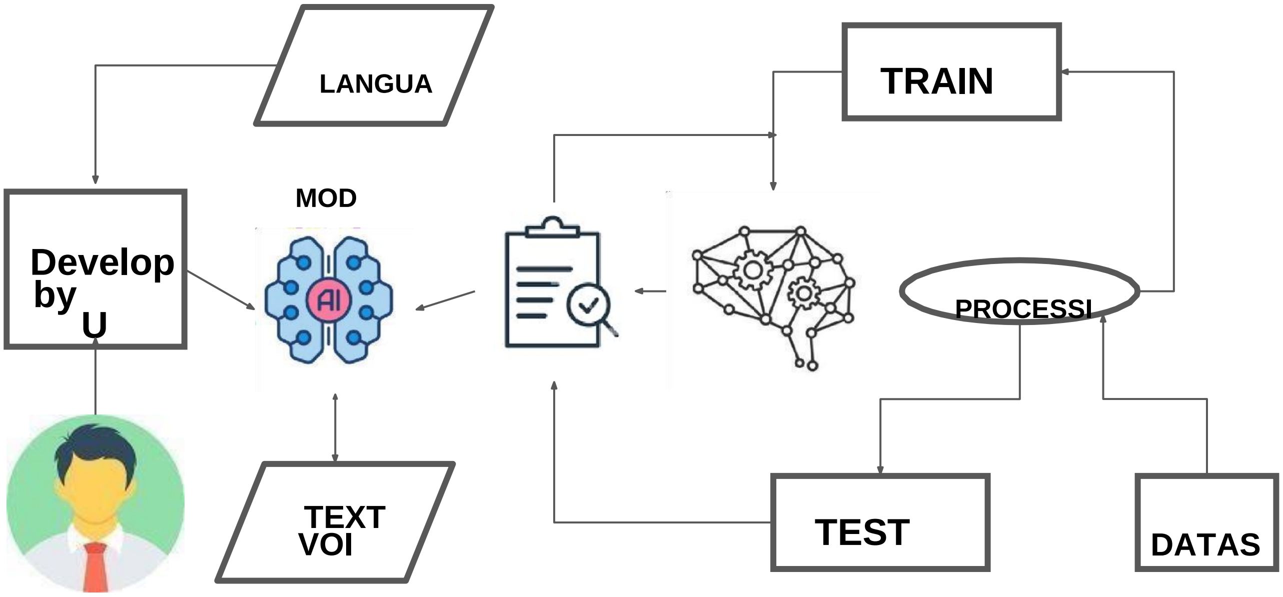


5.2 SOLUTION & TECHNICAL ARCHITECHTURE

SOLUTION



TECHNOLOGY



5.3 USER STORIES

User Type	Functional requirements	User Story Number	User Story/ Tasks	Acceptance criteria	Priority	Release
Customer (psychiatrist)	Registration	USN-1	As a user,I want a sign language translator To fulfill the communication gap between me and my clients	I can access my account/ dashboard	High	Sprint-1
Customer (Teacher)		USN-2	As a user, I want a sign language translator to fill the communication gap between me and my students	I can receive confirmation email & click confirm	High	Sprint-1
Customer (11 Year old boy with profound hearing loss)		USN-3	As a user, I want a system to understand conversations in a better way and respond back	I can register & access the dashboard with facebook login	Low	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Milestone and Activity List

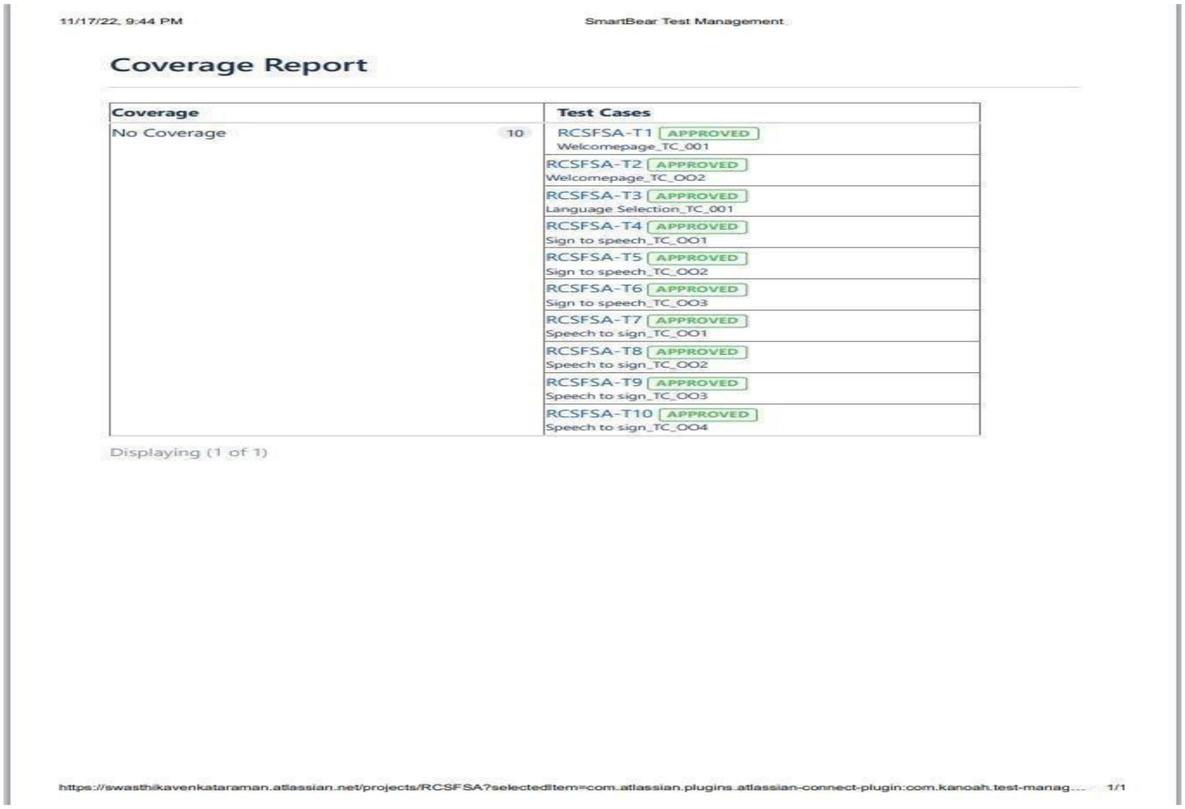
TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	**Literature survey on the selected project & gathering information by referring the, technical papers,research publications etc.**	**10 OCTOBER 2022**
Prepare Empathy Map	**Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements**	**12 OCTOBER 2022**
Ideation	**List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.**	**12 OCTOBER 2022**
Proposed Solution	**Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.**	**12 OCTOBER 2022**
Problem Solution Fit	**Prepare problem - solution fit document.**	**12 OCTOBER 2022**

Solution Architecture	**Prepare solution architecture document.**	**12 OCTOBER 2022**
Customer Journey	**Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).**	**13 OCTOBER 2022**
Data Flow Diagrams	**Draw the data flow diagrams and submit for review.**	**15 OCTOBER 2022**
Technology Architecture	**Prepare the technology architecture diagram.**	**15 OCTOBER 2022**
Prepare Milestone & Activity List	**Prepare the milestones & activity list of the project.**	**25 OCTOBER 2022**
Project Development - Delivery of Sprint-1, 2, 3 & 4	**Develop & submit the developed code by testing it.**	**IN PROGRESS.**

6.2 SPRINT DELIVERY SCHEDULE

Sprint-1	Data Collection	USN-1	Collect Dataset	5	High	Pooja Yuvasri
Sprint1		USN-2	Collect Key points using Media Pipe Holistic	5	High	Akilandeshwari
Sprint2	Model Building	USN-3	Model initialisation with required layers	5	High	Kaviyadharshini
Sprint2		USN-4	Training model using LSTM from key points collected	5	Medium	princy
Sprint3	Testing	USN-5	Testing the model’s performance	10	High	pooja
Sprint4	Speech feature implementa tion	USN-6	Converting text to speech using google API	10	Medium	Yuvasri

6.3 REPORT FROM JIRA



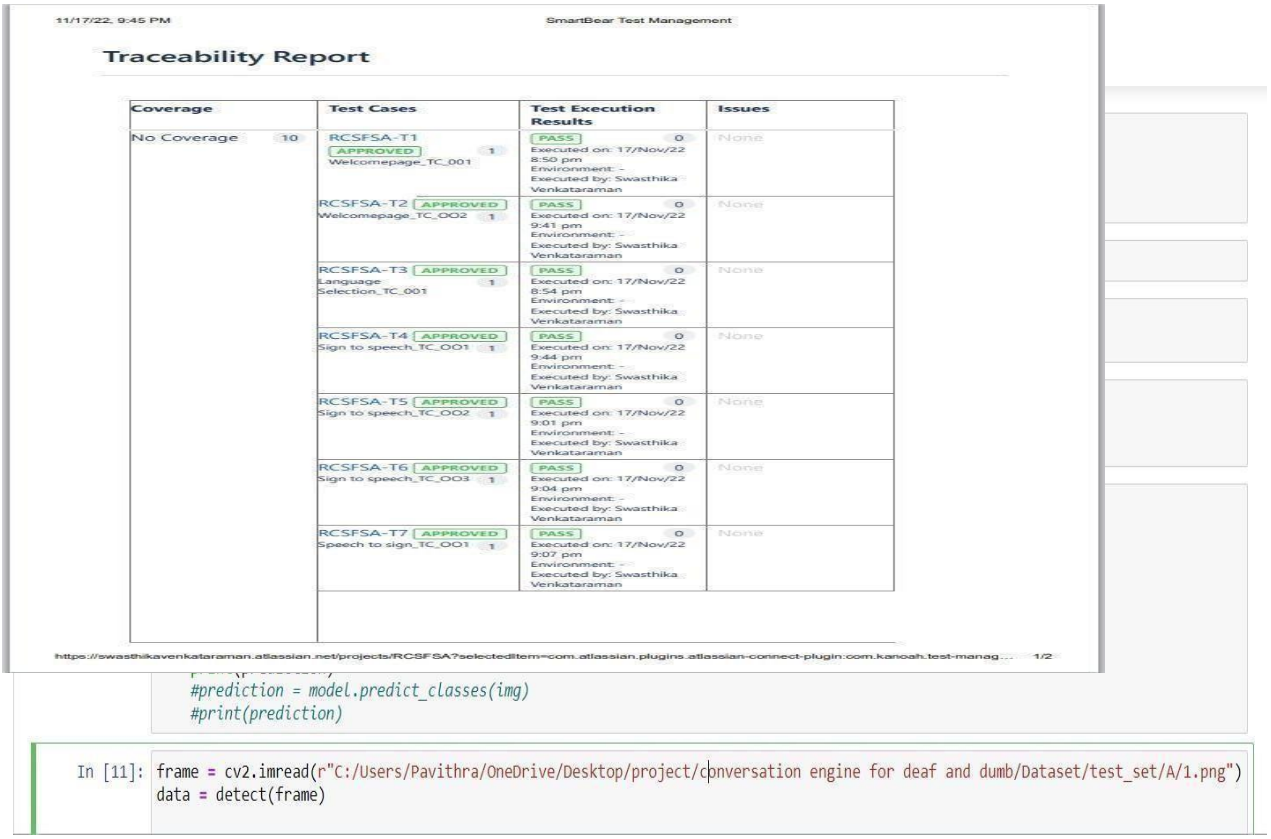
11/17/22, 9:44 PM SmartBear Test Management

Coverage Report

Coverage		Test Cases
No Coverage	10	RCSFSA-T1 APPROVED Welcomepage_TC_001
		RCSFSA-T2 APPROVED Welcomepage_TC_OO2
		RCSFSA-T3 APPROVED Language Selection_TC_001
		RCSFSA-T4 APPROVED Sign to speech_TC_OO1
		RCSFSA-T5 APPROVED Sign to speech_TC_OO2
		RCSFSA-T6 APPROVED Sign to speech_TC_OO3
		RCSFSA-T7 APPROVED Speech to sign_TC_OO1
		RCSFSA-T8 APPROVED Speech to sign_TC_OO2
		RCSFSA-T9 APPROVED Speech to sign_TC_OO3
		RCSFSA-T10 APPROVED Speech to sign_TC_OO4

Displaying (1 of 1)

https://swasthikavenkataraman.atlassian.net/projects/RCSFSA?selectedItem=com.atlassian.plugins.atlassian-connect-plugin:com.kanoah.test-manag... 1/1



11/17/22, 9:45 PM SmartBear Test Management

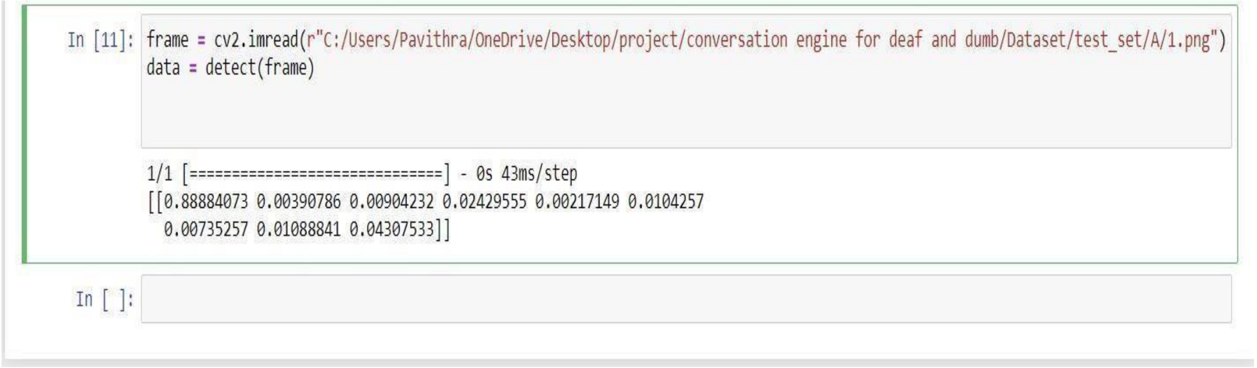
Traceability Report

Coverage	Test Cases	Test Execution Results	Issues
No Coverage 10	RCSFSA-T1 APPROVED 1 Welcomepage_TC_001	PASS 0 Executed on: 17/Nov/22 8:50 pm Environment: - Executed by: Swasthika Venkataraman	None
	RCSFSA-T2 APPROVED Welcomepage_TC_OO2 1	PASS 0 Executed on: 17/Nov/22 9:41 pm Environment: - Executed by: Swasthika Venkataraman	None
	RCSFSA-T3 APPROVED Language Selection_TC_001 1	PASS 0 Executed on: 17/Nov/22 8:54 pm Environment: - Executed by: Swasthika Venkataraman	None
	RCSFSA-T4 APPROVED Sign to speech_TC_OO1 1	PASS 0 Executed on: 17/Nov/22 9:44 pm Environment: - Executed by: Swasthika Venkataraman	None
	RCSFSA-T5 APPROVED Sign to speech_TC_OO2 1	PASS 0 Executed on: 17/Nov/22 9:01 pm Environment: - Executed by: Swasthika Venkataraman	None
	RCSFSA-T6 APPROVED Sign to speech_TC_OO3 1	PASS 0 Executed on: 17/Nov/22 9:04 pm Environment: - Executed by: Swasthika Venkataraman	None
	RCSFSA-T7 APPROVED Speech to sign_TC_OO1 1	PASS 0 Executed on: 17/Nov/22 9:07 pm Environment: - Executed by: Swasthika Venkataraman	None

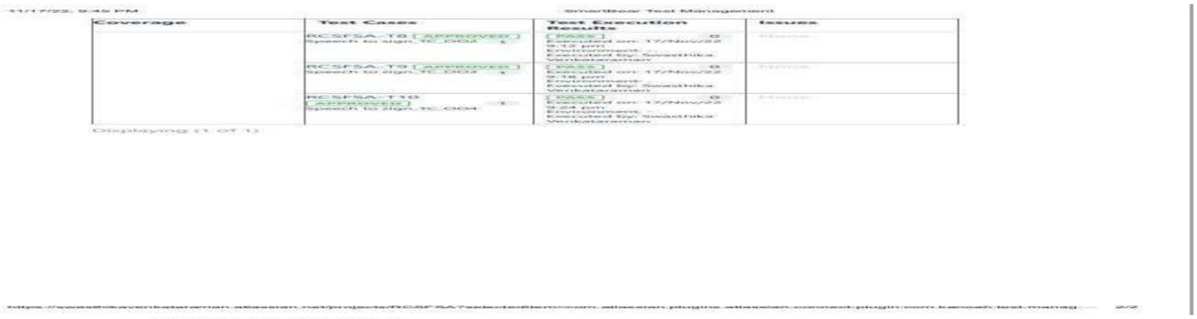
https://swasthikavenkataraman.atlassian.net/projects/RCSFSA?selectedItem=com.atlassian.plugins.atlassian-connect-plugin:com.kanoah.test-manag... 1/2

```
    #prediction = model.predict_classes(img)
    #print(prediction)
```

```
In [11]: frame = cv2.imread(r"C:/Users/Pavithra/OneDrive/Desktop/project/conversation engine for deaf and dumb/Dataset/test_set/A/1.png")
         data = detect(frame)
```



```
In [11]: frame = cv2.imread(r"C:/Users/Pavithra/OneDrive/Desktop/project/conversation engine for deaf and dumb/Dataset/test_set/A/1.png")
         data = detect(frame)

         1/1 [==============================] - 0s 43ms/step
         [[0.88884073 0.00390786 0.00904232 0.02429555 0.00217149 0.0104257
           0.00735257 0.01088841 0.04307533]]

In [ ]:
```

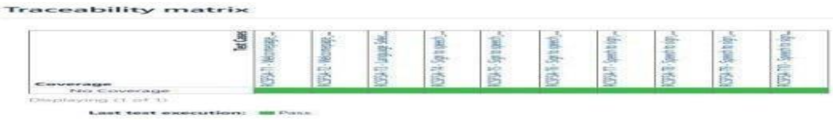


11/17/22, 9:45 PM SmartBear Test Management

Coverage	Test Cases	Test Execution Results	Issues
	RCSFSA-T8 APPROVED Speech to sign_TC_002 1	PASS 0 Executed on: 17/Nov/22 9:12 pm Environment: - Executed by: Swasthika Venkataraman	[illegible]
	RCSFSA-T9 APPROVED Speech to sign_TC_003 1	PASS 0 Executed on: 17/Nov/22 9:18 pm Environment: - Executed by: Swasthika Venkataraman	[illegible]
	RCSFSA-T10 APPROVED Speech to sign_TC_004 1	PASS 0 Executed on: 17/Nov/22 9:24 pm Environment: - Executed by: Swasthika Venkataraman	[illegible]

Displaying (1 of 1)

https://swasthikavenkataraman.atlassian.net/projects/RCSFSA?selectedItem=com.atlassian.plugins.atlassian-connect-plugin:com.kanoah.test-manag... 2/2



Traceability matrix

Test Cases	RCSFSA-T1: Welcome page...	RCSFSA-T2: Welcome page...	RCSFSA-T3: Language Selec...	RCSFSA-T4: Sign to speech...	RCSFSA-T5: Sign to speech...	RCSFSA-T6: Sign to speech...	RCSFSA-T7: Speech to sign...	RCSFSA-T8: Speech to sign...	RCSFSA-T9: Speech to sign...	RCSFSA-T10: Speech to sign...
Coverage										
No Coverage	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

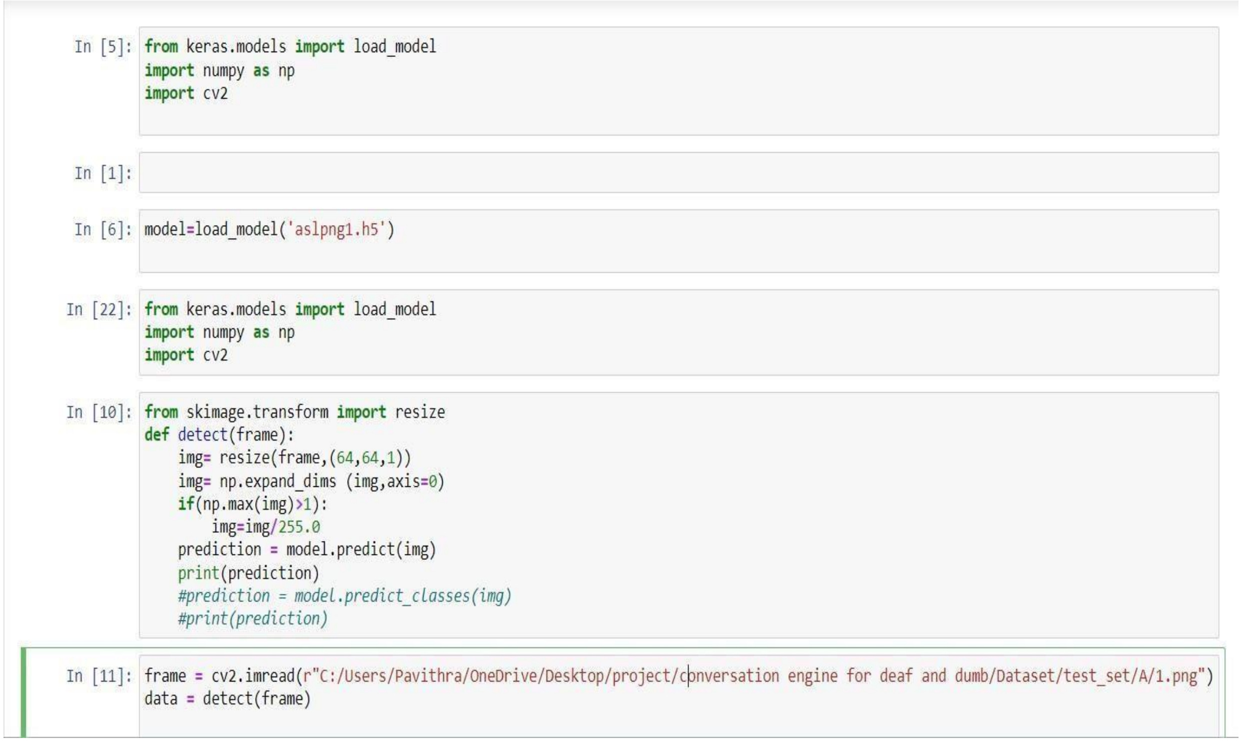
Displaying (1 of 1)

Last test execution: 🟩 Pass

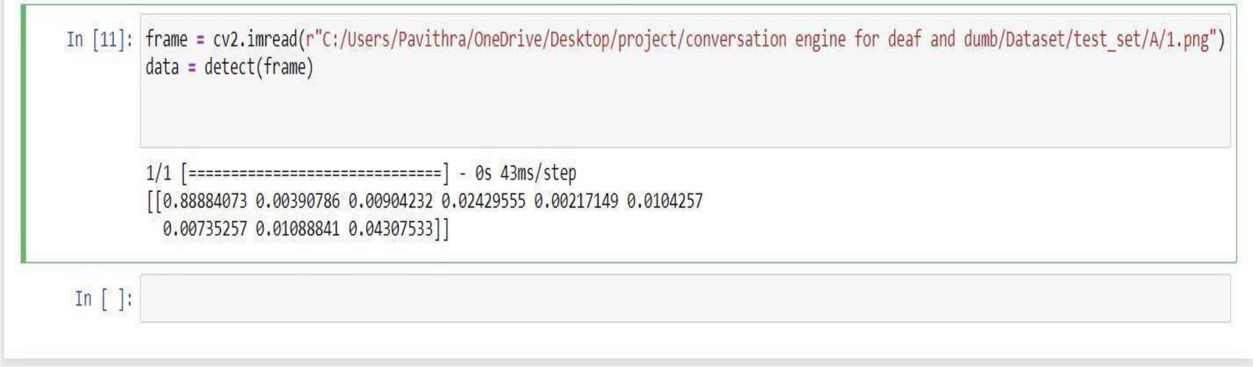
7.CODING & SOLUTIONING

7.1 FEATURE 1

LOAD THE TEST

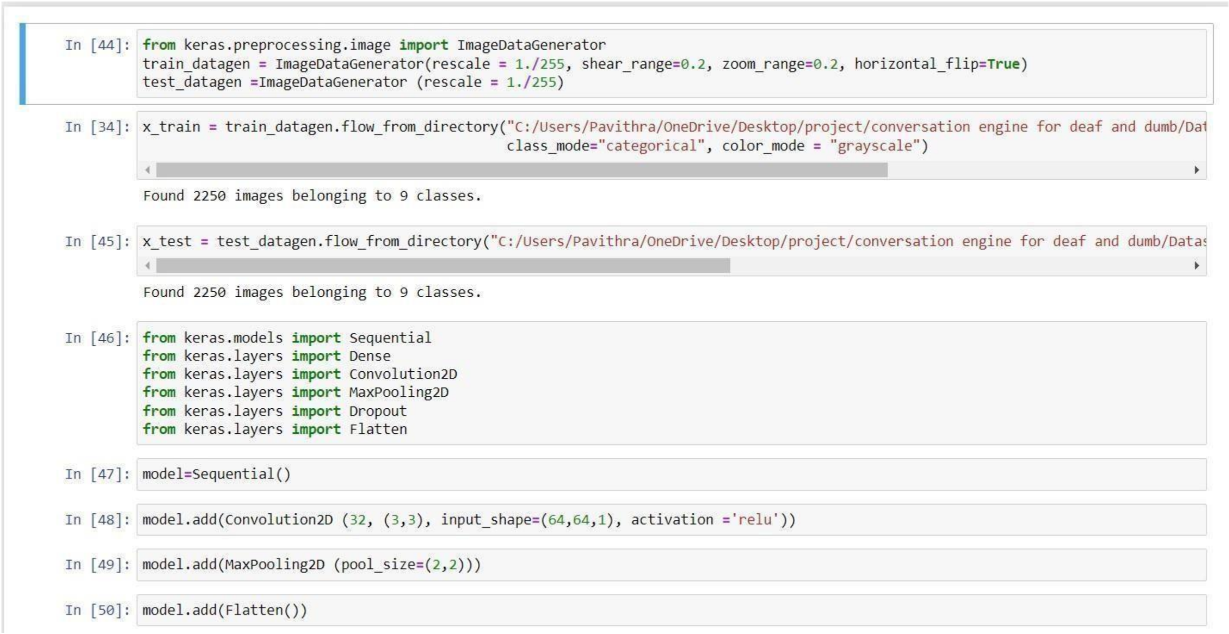


```
In [5]: from keras.models import load_model
        import numpy as np
        import cv2

In [1]:

In [6]: model=load_model('aslpng1.h5')

In [22]: from keras.models import load_model
         import numpy as np
         import cv2

In [10]: from skimage.transform import resize
         def detect(frame):
             img= resize(frame,(64,64,1))
             img= np.expand_dims (img,axis=0)
             if(np.max(img)>1):
                 img=img/255.0
             prediction = model.predict(img)
             print(prediction)
             #prediction = model.predict_classes(img)
             #print(prediction)

In [11]: frame = cv2.imread(r"C:/Users/Pavithra/OneDrive/Desktop/project/conversation engine for deaf and dumb/Dataset/test_set/A/1.png")
         data = detect(frame)
```

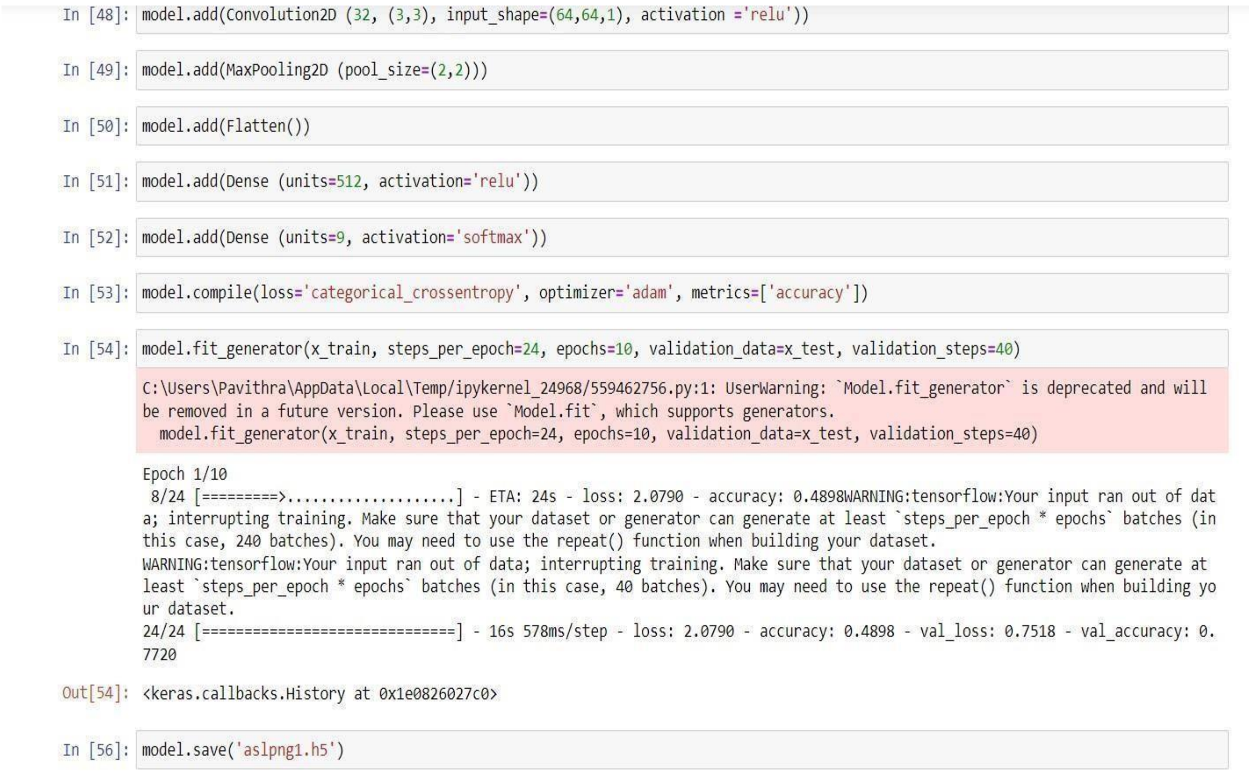


```
In [11]: frame = cv2.imread(r"C:/Users/Pavithra/OneDrive/Desktop/project/conversation engine for deaf and dumb/Dataset/test_set/A/1.png")
         data = detect(frame)

         1/1 [==============================] - 0s 43ms/step
         [[0.88884073 0.00390786 0.00904232 0.02429555 0.00217149 0.0104257
           0.00735257 0.01088841 0.04307533]]

In [ ]:
```

TESTING THE MODEL



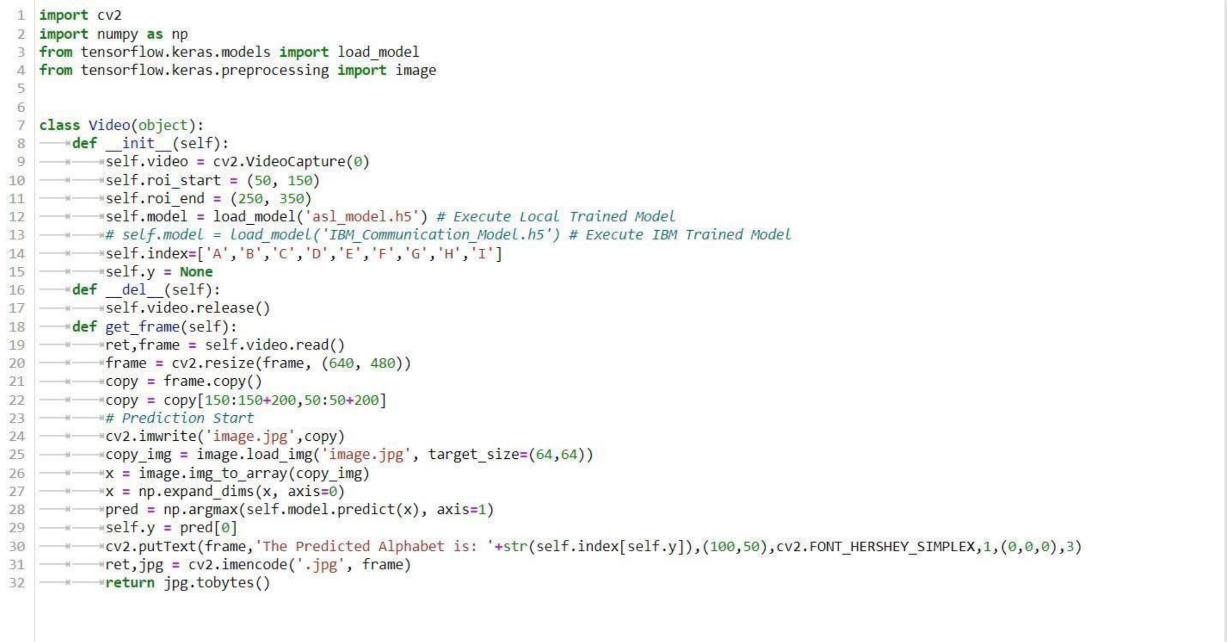
```
In [44]: from keras.preprocessing.image import ImageDataGenerator
         train_datagen = ImageDataGenerator(rescale = 1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
         test_datagen =ImageDataGenerator (rescale = 1./255)

In [34]: x_train = train_datagen.flow_from_directory("C:/Users/Pavithra/OneDrive/Desktop/project/conversation engine for deaf and dumb/Dat
                                                     class_mode="categorical", color_mode = "grayscale")

         Found 2250 images belonging to 9 classes.

In [45]: x_test = test_datagen.flow_from_directory("C:/Users/Pavithra/OneDrive/Desktop/project/conversation engine for deaf and dumb/Datas

         Found 2250 images belonging to 9 classes.

In [46]: from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import Convolution2D
         from keras.layers import MaxPooling2D
         from keras.layers import Dropout
         from keras.layers import Flatten

In [47]: model=Sequential()

In [48]: model.add(Convolution2D (32, (3,3), input_shape=(64,64,1), activation ='relu'))

In [49]: model.add(MaxPooling2D (pool_size=(2,2)))

In [50]: model.add(Flatten())
```



```
In [48]: model.add(Convolution2D (32, (3,3), input_shape=(64,64,1), activation ='relu'))

In [49]: model.add(MaxPooling2D (pool_size=(2,2)))

In [50]: model.add(Flatten())

In [51]: model.add(Dense (units=512, activation='relu'))

In [52]: model.add(Dense (units=9, activation='softmax'))

In [53]: model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

In [54]: model.fit_generator(x_train, steps_per_epoch=24, epochs=10, validation_data=x_test, validation_steps=40)

C:\Users\Pavithra\AppData\Local\Temp/ipykernel_24968/559462756.py:1: 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=24, epochs=10, validation_data=x_test, validation_steps=40)

Epoch 1/10
 8/24 [=========>....................] - ETA: 24s - loss: 2.0790 - accuracy: 0.4898WARNING:tensorflow:Your input ran out of dat
a; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` batches (in
this case, 240 batches). You may need to use the repeat() function when building your dataset.
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at
least `steps_per_epoch * epochs` batches (in this case, 40 batches). You may need to use the repeat() function when building yo
ur dataset.
24/24 [==============================] - 16s 578ms/step - loss: 2.0790 - accuracy: 0.4898 - val_loss: 0.7518 - val_accuracy: 0.
7720

Out[54]: <keras.callbacks.History at 0x1e0826027c0>

In [56]: model.save('aslpng1.h5')
```

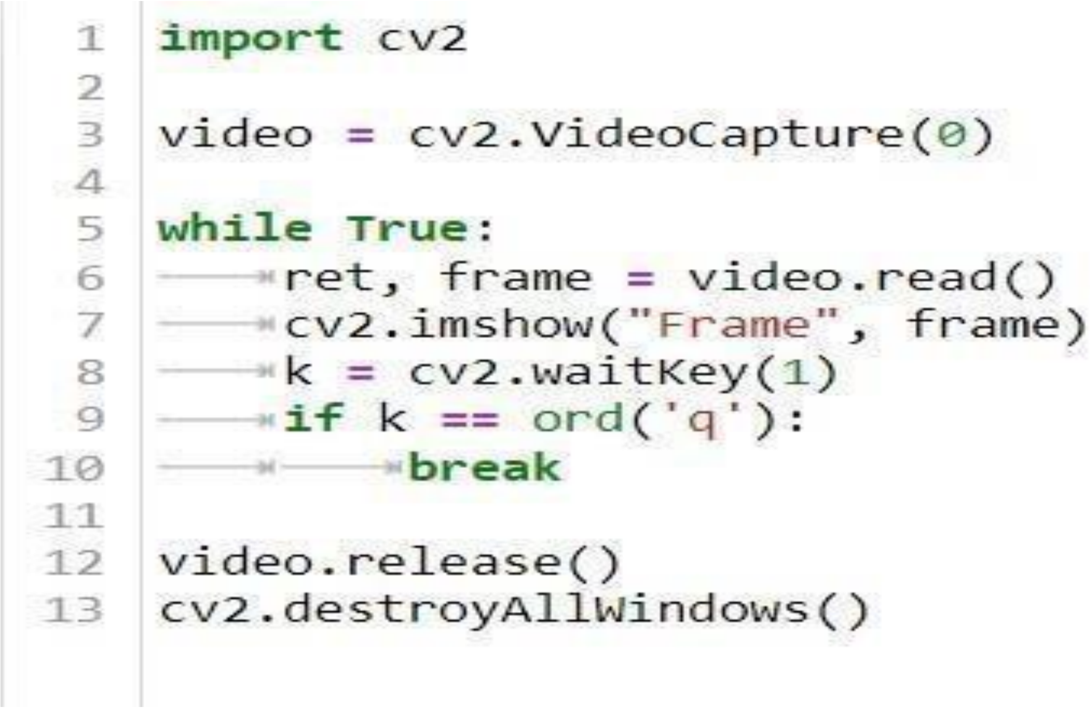
7.2 FEATURE 2

CAMERA.PY



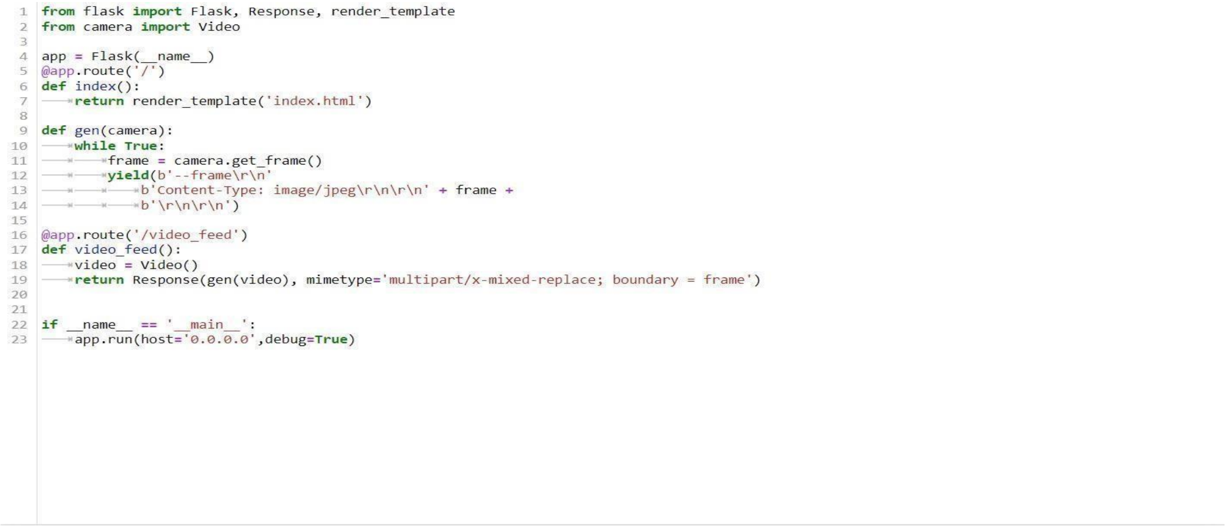
```python
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image


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))
        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()
```

APP.PY



```python
from flask import Flask, Response, render_template
from camera import Video

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

def gen(camera):
    while True:
        frame = camera.get_frame()
        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')


if __name__ == '__main__':
    app.run(host='0.0.0.0',debug=True)
```

1. **Generation of the database.**

Here our system takes the hand movements through the web camera. In this proposed method, 26 combinations of Indian characters are developed by the use of right Hand saved in training database.

2. **Image preprocessing and segmentation.**

The pre-processing takes place on these recorded input gestures. Then the segmentation Hands are performed to separate object and background

3. **Feature Extraction.**

The segmented hand image is represented with certain features. The characteristics are used for gesture recognition with the template matching algorithm that gives Optimized results.

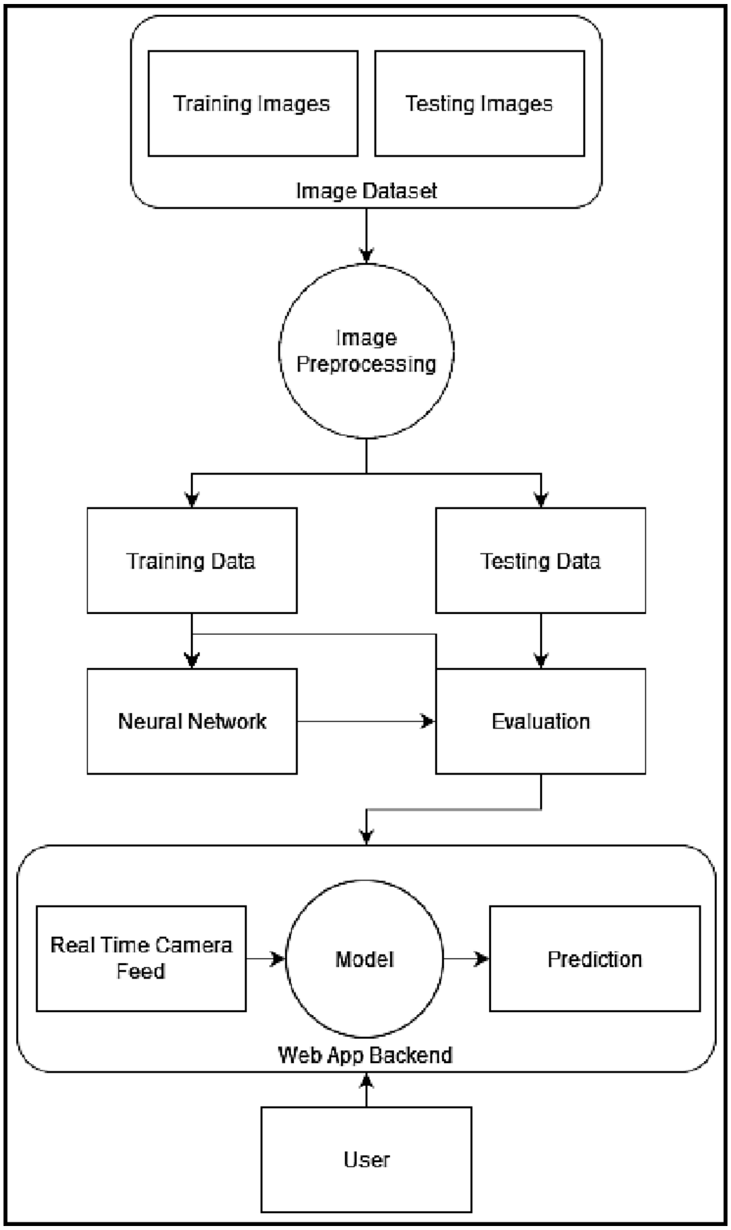
4. **Sign recognition.**

The given character gesture is recognized with the skin color recognition and the template Matching from the record.

5. **Sign to text and Speech conversion.**

The recognized sign is then mapped into text and further converted into speech With TTS libraries.

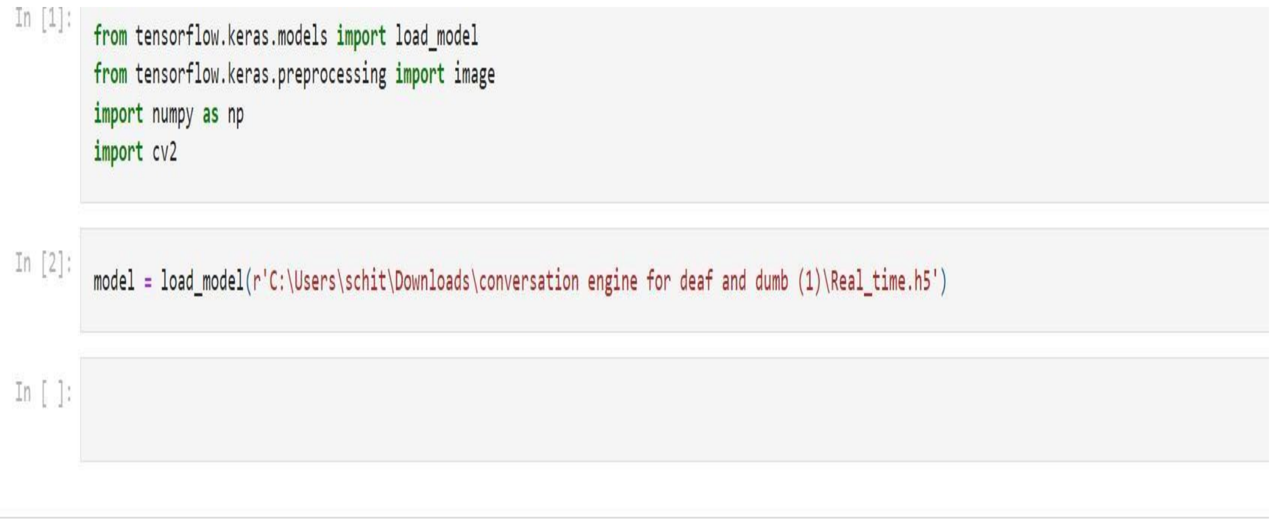
7.3 DATABASE SCHEMA



8. TESTING

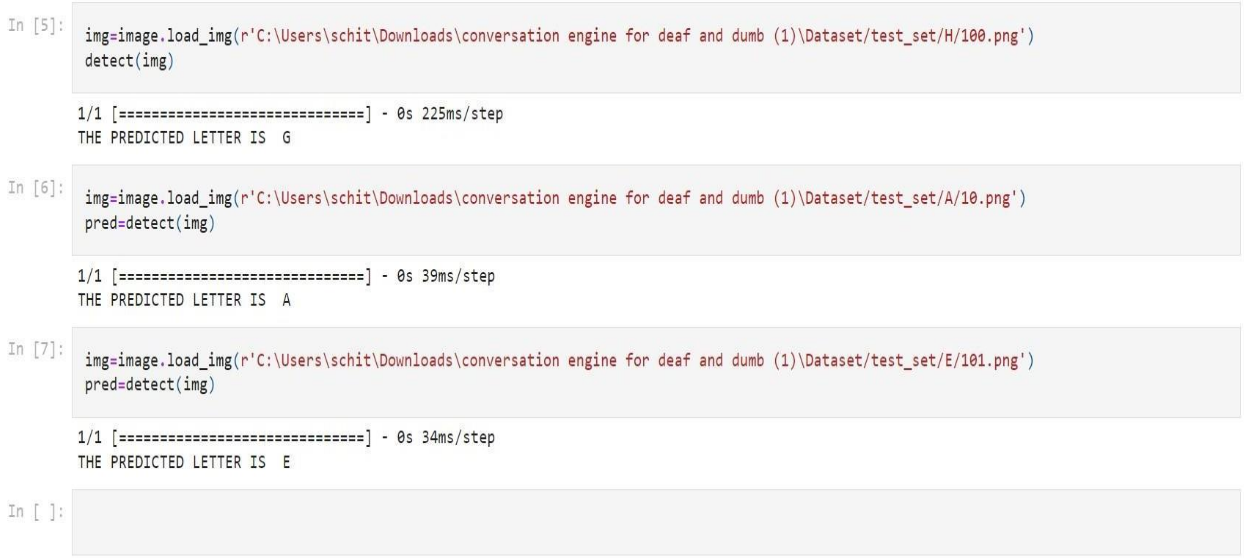
8.1 TEST CASES

Import the packages and load the saved model.ipynb



```
In [1]:  from tensorflow.keras.models import load_model
         from tensorflow.keras.preprocessing import image
         import numpy as np
         import cv2

In [2]:  model = load_model(r'C:\Users\schit\Downloads\conversation engine for deaf and dumb (1)\Real_time.h5')

In [ ]:
```

Load the test image pre-proces it and predict.ipynb



```
In [5]:  img=image.load_img(r'C:\Users\schit\Downloads\conversation engine for deaf and dumb (1)\Dataset/test_set/H/100.png')
         detect(img)

         1/1 [==============================] - 0s 225ms/step
         THE PREDICTED LETTER IS  G

In [6]:  img=image.load_img(r'C:\Users\schit\Downloads\conversation engine for deaf and dumb (1)\Dataset/test_set/A/10.png')
         pred=detect(img)

         1/1 [==============================] - 0s 39ms/step
         THE PREDICTED LETTER IS  A

In [7]:  img=image.load_img(r'C:\Users\schit\Downloads\conversation engine for deaf and dumb (1)\Dataset/test_set/E/101.png')
         pred=detect(img)

         1/1 [==============================] - 0s 34ms/step
         THE PREDICTED LETTER IS  E

In [ ]:
```

8.2 USER ACCEPTANCE TESTING

1. DEFECT ANALYSYS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	11	7	4	2	24
Duplicate	1	0	2	0	3
External	2	3	2	1	8
Fixed	10	5	3	14	32
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	1	0	0	0	1
Totals	25	15	13	18	71

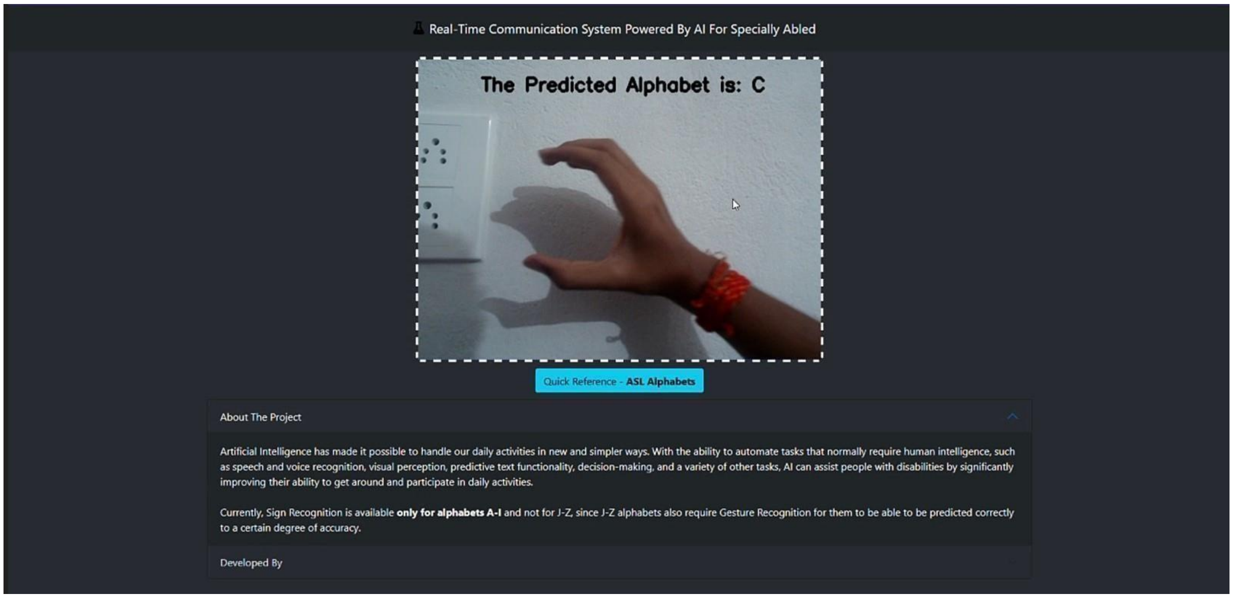
2. TESTCASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	15	0	0	15
Security	2	0	0	2
Outsource Shipping	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 PERFORMANCE METRICS

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:



Real-Time Communication System Powered By AI For Specially Abled

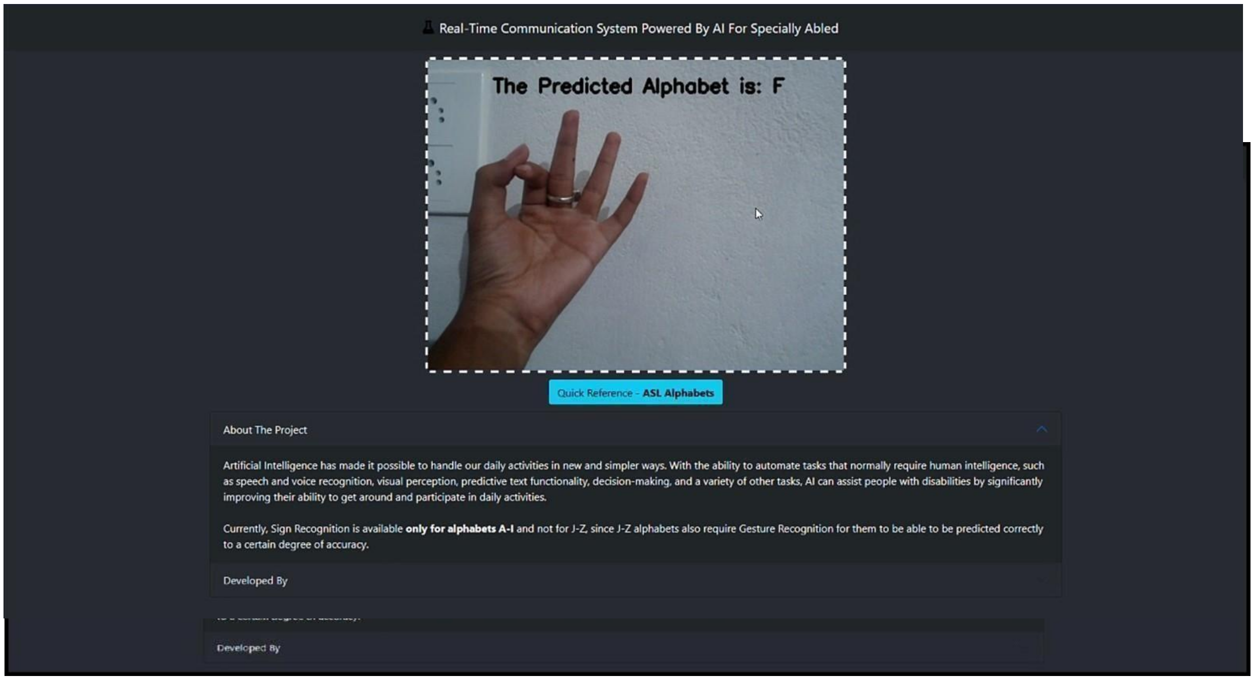
The Predicted Alphabet is: C

About The Project

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.

Currently, Sign Recognition is available **only for alphabets A-I** 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.

Developed By


Real-Time Communication System Powered By AI For Specially Abled
The Predicted Alphabet is: F
Quick Reference - ASL Alphabets

About The Project

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.

Currently, Sign Recognition is available **only for alphabets A-I** 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.

Developed By

Developed By

10. ADVANTAGES & DISADVANTAGE

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 interfaces.

13. APPENDEX

SOURCE CODE

STATIC

CSS

Banner-Heading-Image.css

```
.fit-cover {
  object-fit: cover;
}
```

Navbar-Centered-Brand.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;
}
```

TEMPLATES

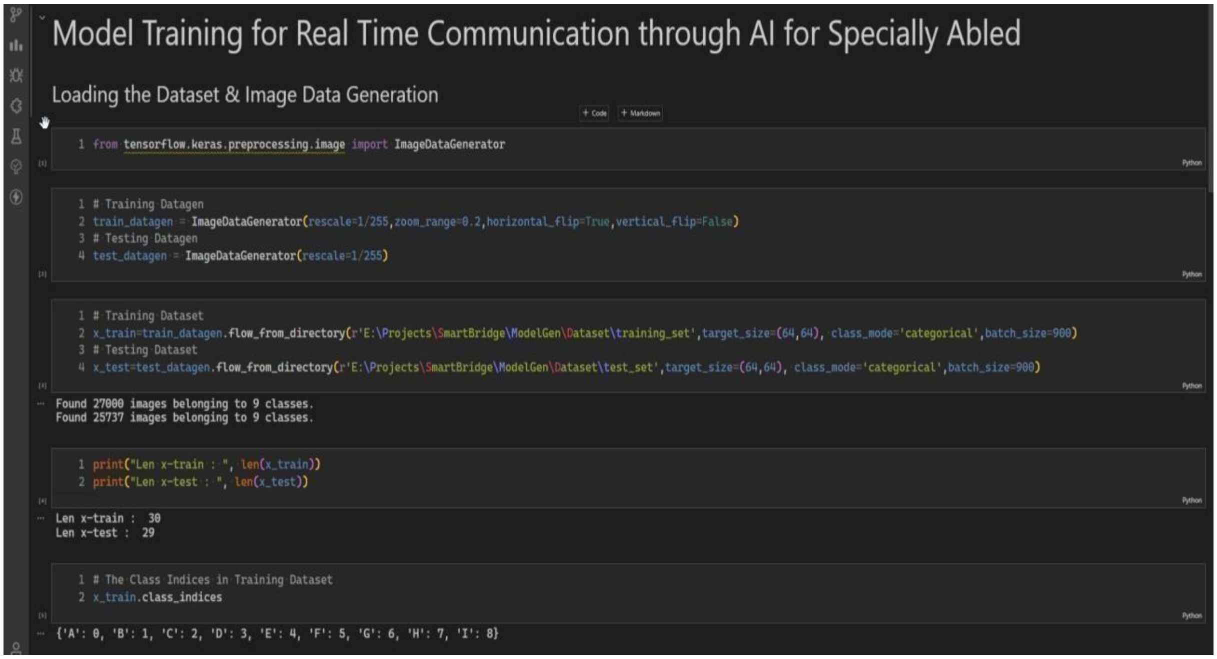
INDEX.HTML

```
<!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">
</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"
              aria-controls="accordion-1 .item-1"
              style="background: rgb(39,43,48);color: rgb(255,255,255);">About The 
Project</button></h2>
          <div class="accordion-collapse collapse show item-1" role="tabpanel" data-bs-
parent="#accordion-1">
            <div class="accordion-body">
              <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>s.pooja</strong> 312819205028<br>2.
                <strong>s.yuvasri</strong>312819205050<br>3.
<strong>s.kaviyadharshini</strong> 312819205012<br>4.<strong>f.princy</strong>
312819205033<br>5.<strong>A.akilandeswari</strong> 312819205005<br>

              </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>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>

</html>
```

APP.PY

```
#!/usr/bin/env python
# coding: utf-8

# In[13]:



from flask import Flask, Response, render_template
from camera import Video

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

def gen(camera):
while True:
    frame = camera.get_frame()
    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')


if __name___== '_main_':
app.run
# In[ ]:
```

SOURCE CODE FOR MODEL TRAINING AND SAVING



Model Training for Real Time Communication through AI for Specially Abled

Loading the Dataset & Image Data Generation

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

```
# 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)
```

```
# 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.
```

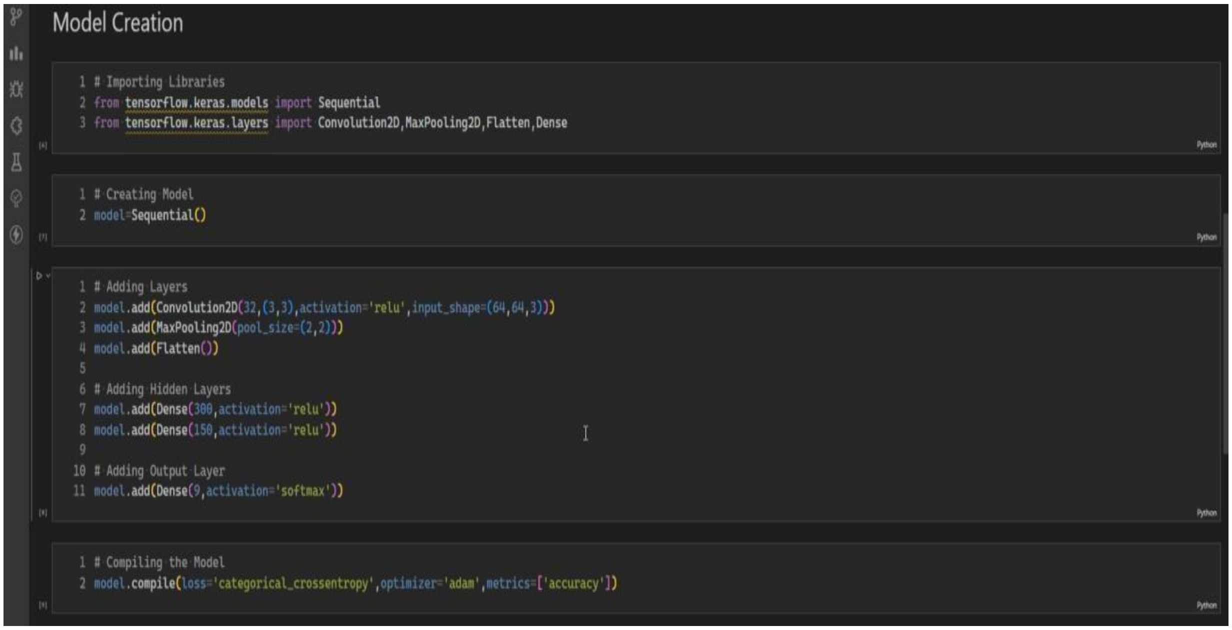
```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

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

```
# 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



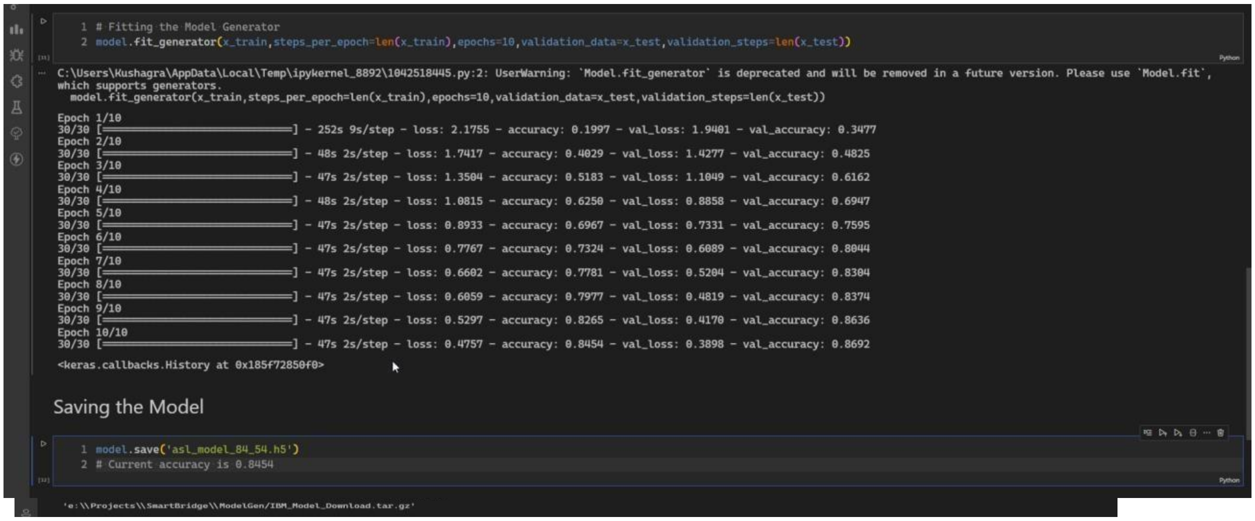
Model Creation

```
# 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'])
```



```python
# 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))
```

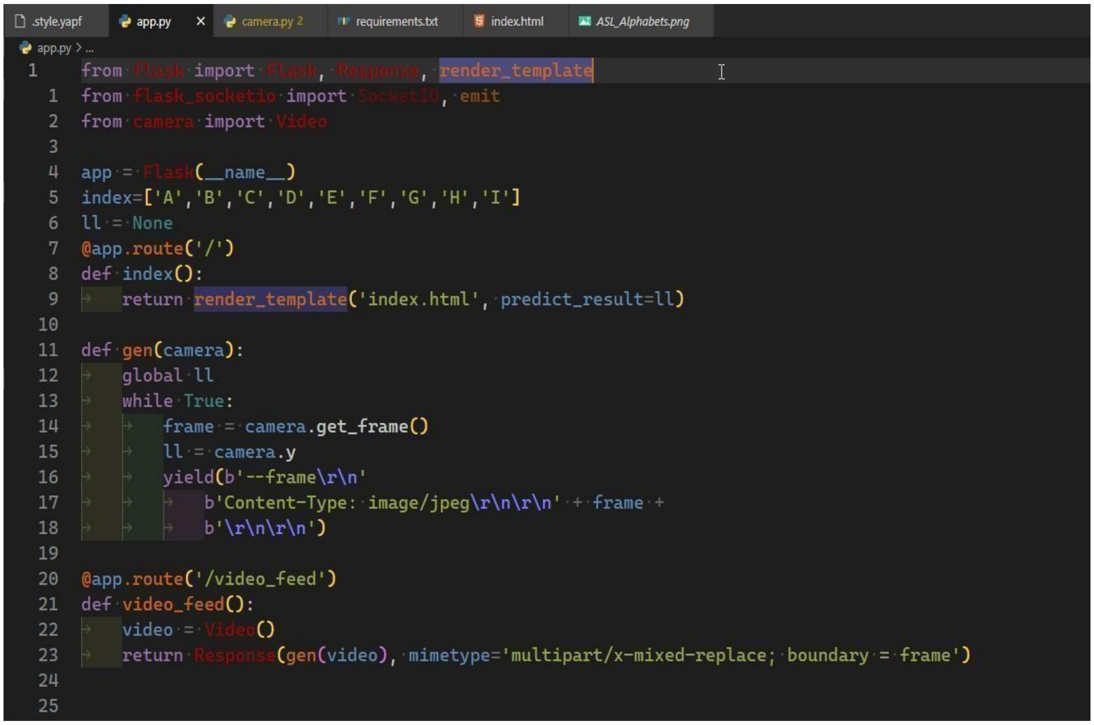
```
C:\Users\Kushagra\AppData\Local\Temp\ipykernel_8892\1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`,
which supports generators.
  model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))

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

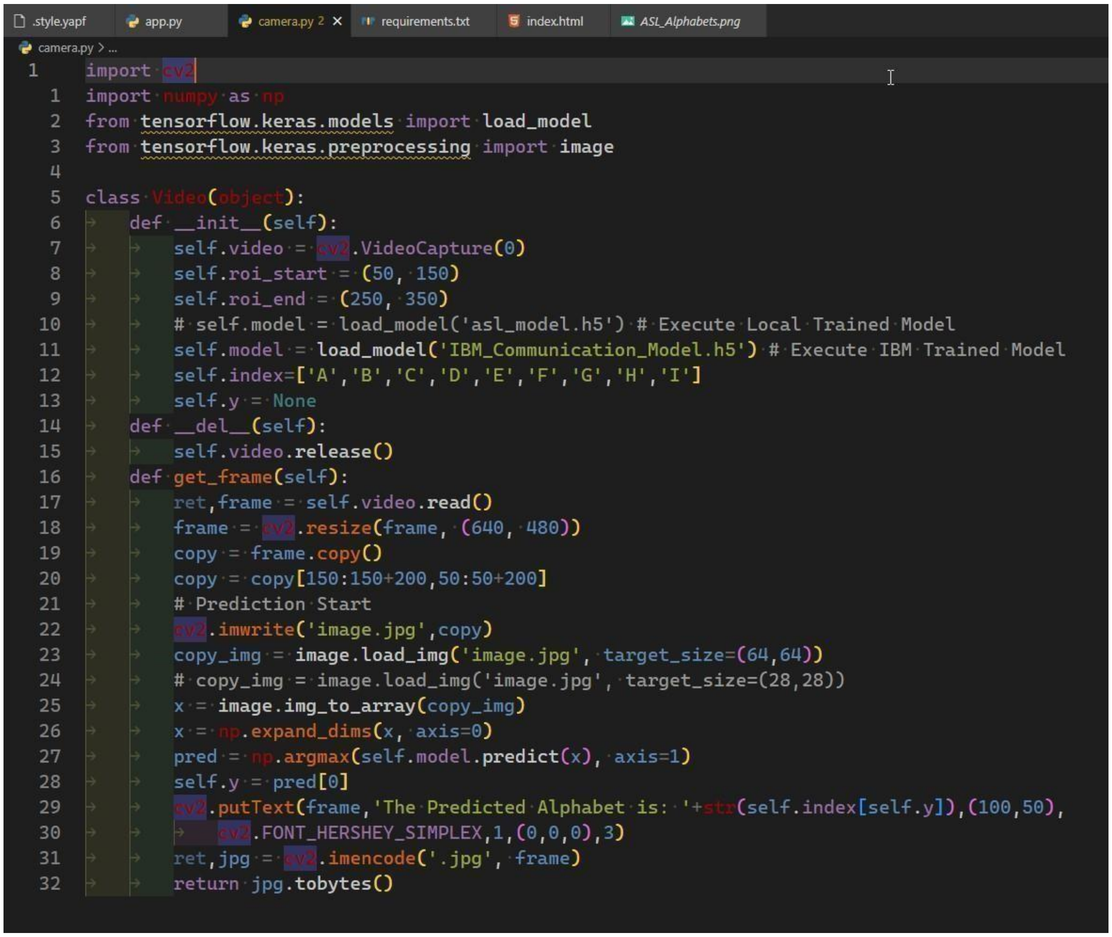
```python
model.save('asl_model_84_54.h5')
# Current accuracy is 0.8454
```

```
'e:\\Projects\\SmartBridge\\ModelGen/IBM_Model_Download.tar.gz'
```

WEB APP CODE

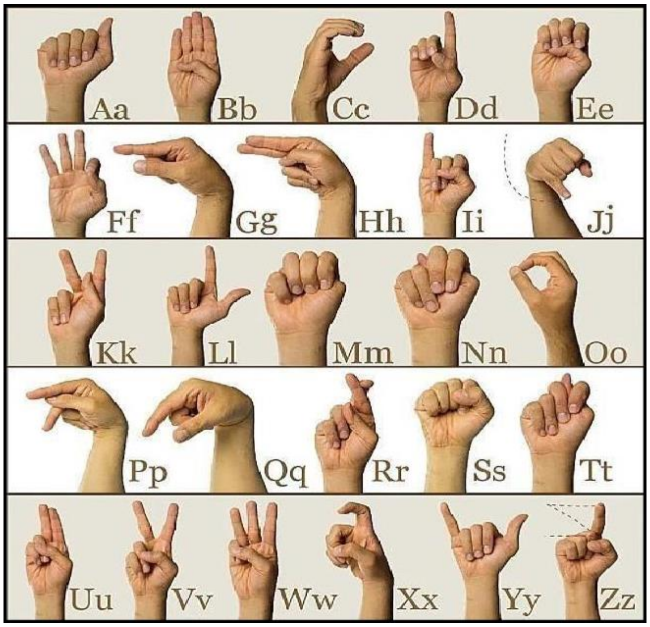


```python
from Flask import Flask, Response, render_template
from flask_socketio import SocketIO, emit
from camera import Video

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')
```



```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 STANDERED REFERENCE



GITHUB AND DEMO ID:

https://github.com/IBM-EPBL/IBM-Project-27574-1660060070