PROJECT REPORT

Project Title : Real-Time Communication System Powered By AI for Specially Abled

Project Members :-

Siddarth V S

Mohanraam V K

Rupesh Kanna S

Sharan Kumar J A S

Harish Kumar S

1) INTRODUCTION

1.1) PROJECT OVERVIEW

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

1.2) PURPOSE OF THE PROJECT

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

There are handicapped people in our society. Although technology is constantly evolving, little is being done to improve the lives of these people. It has always been difficult to communicate with someone who is deaf-mute. It is quite challenging for silent persons to communicate with non-mute people because hand sign language is not taught to the general public. It might be quite challenging for them to communicate at times of crisis. In circumstances where other modes of communication, like speech, are not possible, the human hand has remained a common alternative for information transmission. To have a proper communication between a normal person and a handicapped person in any language, a voice conversion system with hand gesture recognition and translation will be very helpful.

2.2) REFERENCES

[1] Thoutam, N., Jha, D. K., Jaiswal, L., Deshmukh, S., & Raj, R. Hand gesture, Text and Speech Translation and Recognition System for specially abled people using AI.

[2] Shinde, Shweta S., Rajesh M. Autee, and Vitthal K. Bhosale. "Real time two way communication approach for hearing impaired and dumb person based on image processing." Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference on. IEEE, 2016.

[3] Shangeetha, R. K., V. Valliammai, and S. Padmavathi. "Computer vision based approach for Indian Sign Language character recognition." Machine Vision and Image Processing (MVIP), 2012 International Conference on. IEEE, 2012.

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

[5] Ahire, Prashant G., et al. "Two Way Communicator between Deaf and Dumb People and Normal People." Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on. IEEE, 2015.

[6] Ms R. Vinitha and Ms A. Theerthana. "Design And Development Of Hand Gesture Recognition System For Speech Impaired People."

[7] Kusurnika Krori Dutta, Satheesh Kumar Raju K, Anil Kumar G S,Sunny Arokia Swarny B,“Double handed Indian Sign Language to speech and text”, IEEE, 2015 Third International Conference on ImageInformation Processing.

[8] Aarthi M, Vijayalakshmi P, “Sign language to speech conversion”,IEEE,2016 fifth international conference on recent trends in information technology.

[9] ‘‘Malay Sign Language Gesture Recognition System ”Tan TianSwee, Sh-Hussain Salleh, A.K.Ariff, Chee-Ming Ting, Siew KeanSeng, and Leong Seng Huat, ‘‘International Conference on Intelligent And Advanced Systems 2007”

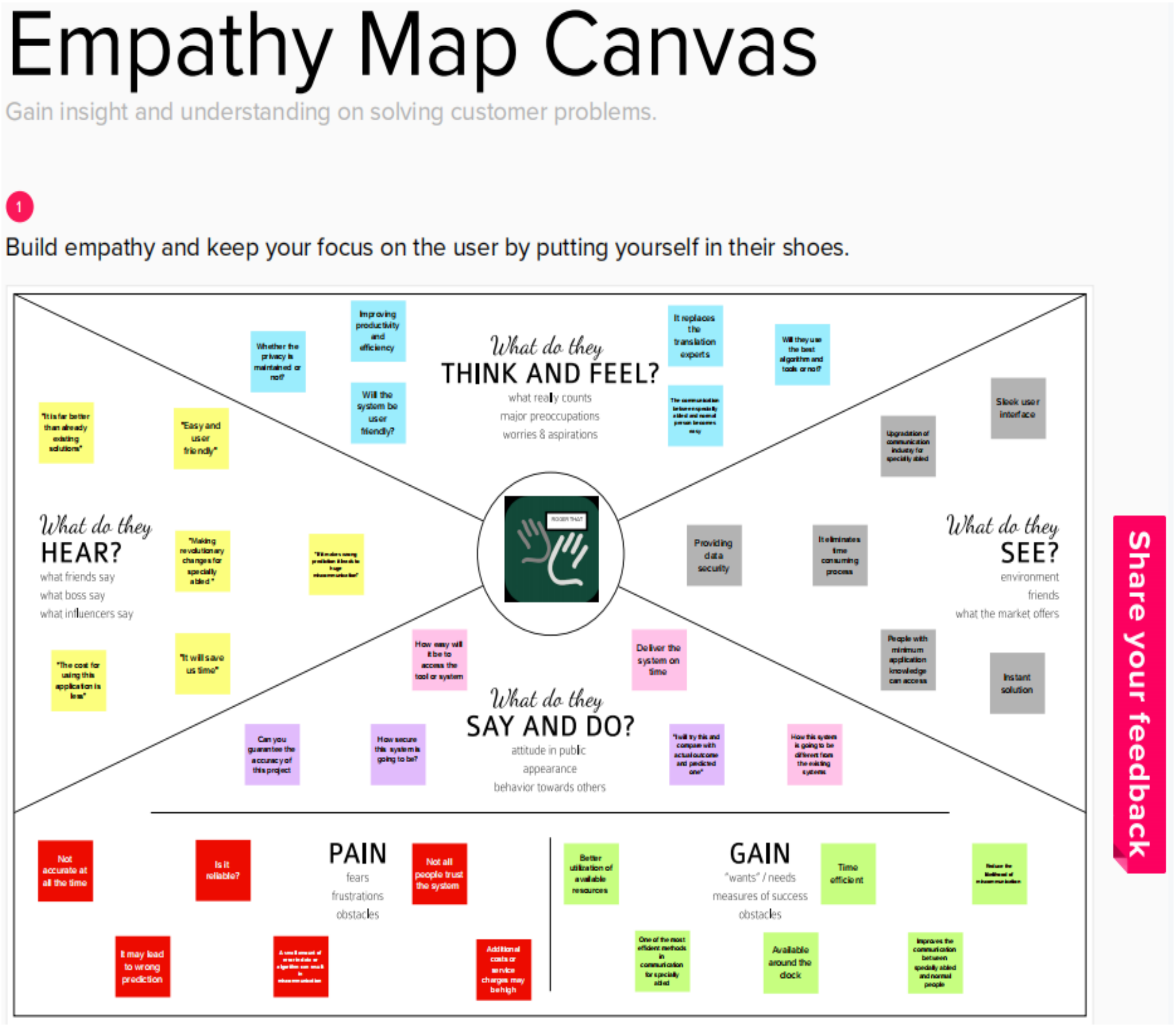
[10] Sarth Pham The Hai, Huynh Chau Thinh, Bui Van Phuc, Ha HoangKha, "Automatic feature extraction for Vietnamese sign language recognition using support vector machine", Recent Advances in Signal Processing Communications & Computing (SigTelCom)2018 2nd International Conference on, pp. 146-151, 2018.

2.3) PROBLEM STATEMENT DEFINITION

Communication is the medium by which we can share our thoughts or convey the messages with other person. But for specially abled people like dumb and deaf people this is challenging. Gesture shows an expressive movement of body which convey some message but not everyone is aware of sign language. Since deaf and dumb people have to depend on some sort of visual communication. In recent years, there has been so many advancement in technology. Gesture recognition is the mathematical interpretation of a human motion by a computing device. Sign language provide best communication platform for the hearing impaired and dumb person to communicate with normal person.

3) IDEATION AND PROPOSED SOLTION

3.1) EMPATHY MAP CANVAS



Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

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

Whether the privacy is maintained or not?

Improving productivity and efficiency

Will the system be user friendly?

It replaces the translation experts

The communication between specially abled and normal person becomes easy

Will they use the best algorithm and tools or not?

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

"It is far better than already existing solutions"

"Easy and user friendly"

"Making revolutionary changes for specially abled"

"If it makes wrong prediction it leads to huge miscommunication"

"The cost for using this application is less"

"It will save us time"

What do they SEE?
environment
friends
what the market offers

Upgradation of communication industry for specially abled

Sleek user interface

Providing data security

It eliminates time consuming process

People with minimum application knowledge can access

Instant solution

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

How easy will it be to access the tool or system

Deliver the system on time

Can you guarantee the accuracy of this project

How secure this system is going to be?

"I will try this and compare with actual outcome and predicted one"

How this system is going to be different from the existing systems

PAIN
fears
frustrations
obstacles

Not accurate at all the time

Is it reliable?

Not all people trust the system

It may lead to wrong prediction

A small amount of error in data or algorithm can result in miscommunication

Additional costs or service charges may be high

GAIN
"wants" / needs
measures of success
obstacles

Better utilization of available resources

Time efficient

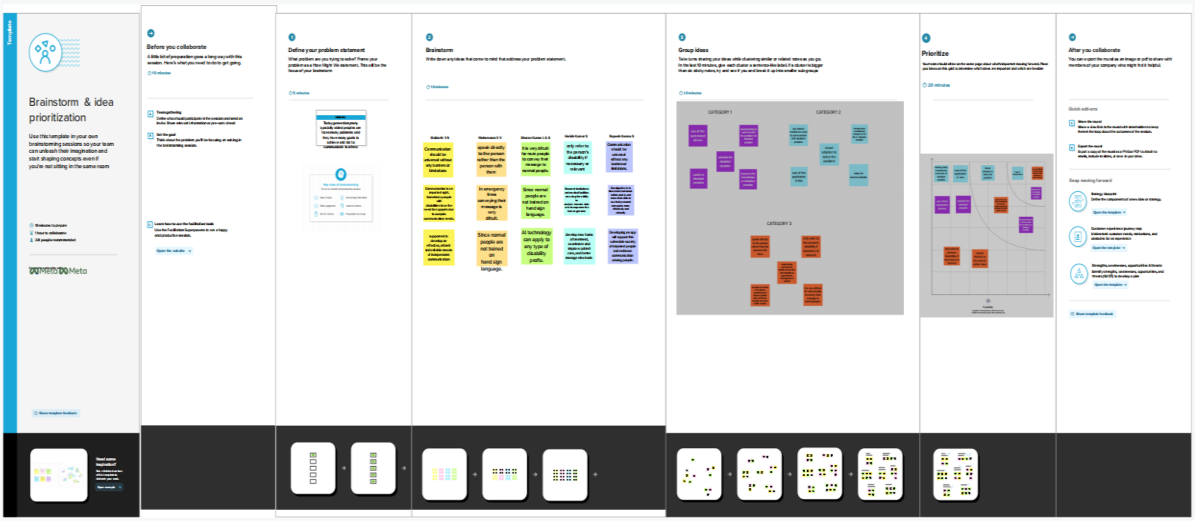
Reduce the likelihood of miscommunication

One of the most efficient methods in communication for specially abled

Available around the clock

Improves the communication between specially abled and normal people

3.2) IDEATION AND 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

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

Team gathering

Set the goal

Learn how to use the facilitation tools

Open the website →

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

2 Brainstorm

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

10 minutes

Communication should be universal without any barriers or limitations

speak directly to the person rather than the person with them

It is very difficult for mute people to convey their message to normal people.

only refer to the person's disability if necessary or relevant

Communication should be universal without any barriers or limitations.

In emergency times conveying their message is very dificult.

Since normal people are not trained on hand sign language.

Since normal people are not trained on hand sign language.

AI technology can apply to any type of disability profile.

Developing an app will support the vulnerable society of impaired people and enhance communication among people.

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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

CATEGORY 1

CATEGORY 2

CATEGORY 3

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

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural

Export the mural

Keep moving forward

Strategy blueprint

Open the template →

Customer experience journey map

Open the template →

Strengths, weaknesses, opportunities & threats

Open the template →

Share template feedback

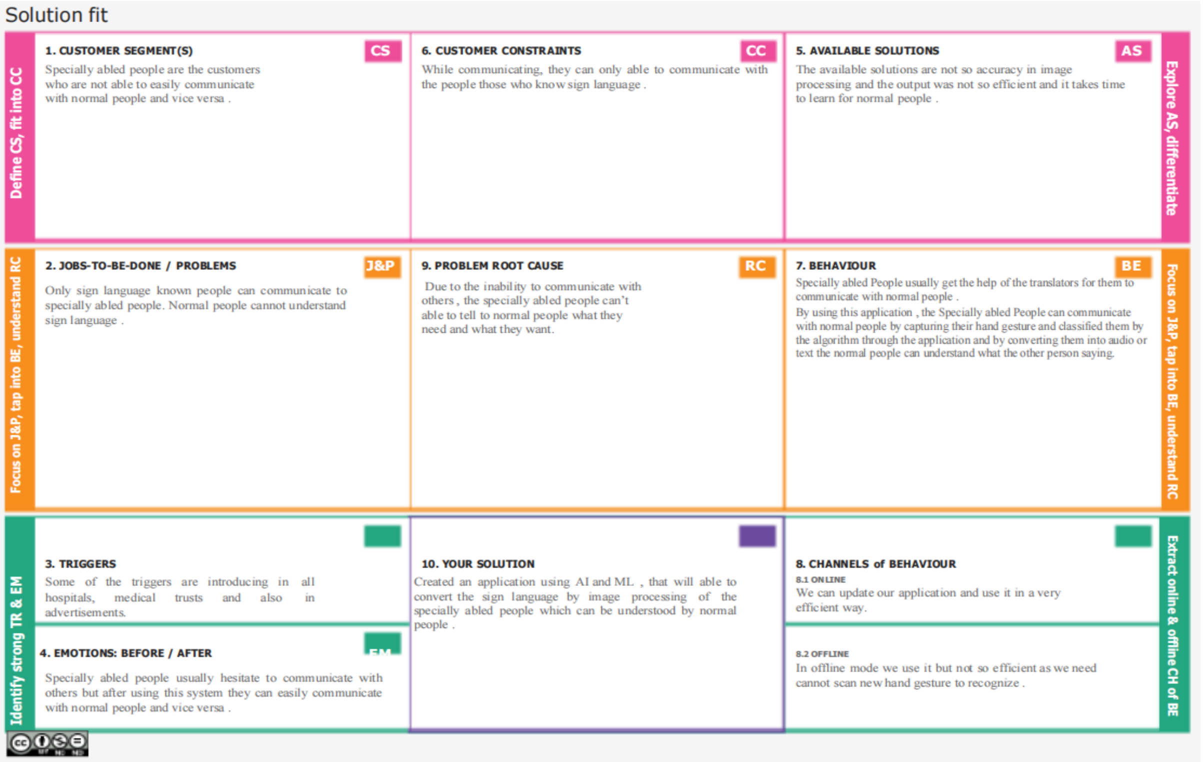
3.3) PROPOSED SOLUTION

Proposed Solution Template:

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To design a application for the deaf and dumb to communicate with the normal people and vice versa .
2.	Idea / Solution description	By analysing the given input sign language pictures(Hand gestures) using AI and ML and then convert them into audio or text .
3.	Novelty / Uniqueness	Two-way communication between normal person and specially abled person made easier . This method has improved efficiency than the others .
4.	Social Impact / Customer Satisfaction	This model makes life easier for the specially abled people with their communication . They can feel free to communicate to normal people than before .
5.	Business Model (Revenue Model)	This app can be released to people in the world, And the application will be installed by all , so it will produce a revenue .
6.	Scalability of the Solution	This solution will bring a new evolution for the specially abled people in communicating with other people .

3.4) PROBLEM SOLUTION FIT


Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Extract online & offline CH of BE
CS
CC
AS
J&P
RC
BE
EM

Solution fit

1. CUSTOMER SEGMENT(S)

Specially abled people are the customers who are not able to easily communicate with normal people and vice versa .

6. CUSTOMER CONSTRAINTS

While communicating, they can only able to communicate with the people those who know sign language .

5. AVAILABLE SOLUTIONS

The available solutions are not so accuracy in image processing and the output was not so efficient and it takes time to learn for normal people .

2. JOBS-TO-BE-DONE / PROBLEMS

Only sign language known people can communicate to specially abled people. Normal people cannot understand sign language .

9. PROBLEM ROOT CAUSE

Due to the inability to communicate with others , the specially abled people can't able to tell to normal people what they need and what they want.

7. BEHAVIOUR

Specially abled People usually get the help of the translators for them to communicate with normal people .

By using this application , the Specially abled People can communicate with normal people by capturing their hand gesture and classified them by the algorithm through the application and by converting them into audio or text the normal people can understand what the other person saying.

3. TRIGGERS

Some of the triggers are introducing in all hospitals, medical trusts and also in advertisements.

4. EMOTIONS: BEFORE / AFTER

Specially abled people usually hesitate to communicate with others but after using this system they can easily communicate with normal people and vice versa .

10. YOUR SOLUTION

Created an application using AI and ML , that will able to convert the sign language by image processing of the specially abled people which can be understood by normal people .

8. CHANNELS of BEHAVIOUR

8.1 ONLINE

We can update our application and use it in a very efficient way.

8.2 OFFLINE

In offline mode we use it but not so efficient as we need cannot scan new hand gesture to recognize .

4) REQUIREMENT ANALYSIS

4.1) FUNCTIONAL REQUIREMENTS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Phone number Registration through Gmail Registration through Facebook ID
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Verification	The user should receive a verification e-mail or OTP which they have to enter it to finish the registration.
FR-4	Device Permission	Permission request for camera access to record real time gestures and to show sign-language output. Permission to store app related files in storage.
FR-5	Tutorial	The user will have a tutorial how to use the app and they will gain knowledge about the app easily.
FR-6	User input through app whether to start scan gesture recognition	Starts capturing video after this option is clicked .
FR-7	User input through app whether to stop the process	Stop Capturing /listening the scan recognition.
FR-8	Save gesture through app	Users can get the option to Save the Gesture inputs in their Device.
FR-9	Providing option to change language of the app in the settings menu	The UI will get reset to the custom language selected by the user.

4.2) NON FUNCTIONAL REQUIREMENTS

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application should be designed in such a way that is easy to use for specially abled persons and it should be portable and platform-independent.
NFR-2	**Security**	The system or the designed application will have Two -factor authentication and the users will have to enter a strong password at the time of registration.
NFR-3	**Reliability**	The system is tested with more samples of data and tested properly with more workloads to ensure that this app is working under all conditions to maintain the service and the reliability of the system.
NFR-4	**Performance**	The system is tested and trained to maintain fast performance , availability and user experience. It measures how fast or slows your app loads if the app crashes when it reaches a peak in user activity, and how smooth certain features (like Camera and Video) work.
NFR-5	**Availability**	The system is available in all platforms . It is available in android , IOS and Computers etc.
NFR-6	**Scalability**	The system has the capacity to handle growth, especially in handling more users and evolving concurrently with your business needs and in providing data on time to the customer.

5) PROJECT DESIGN

5.1) DATA FLOW DIAGRAMS

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 rightamount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

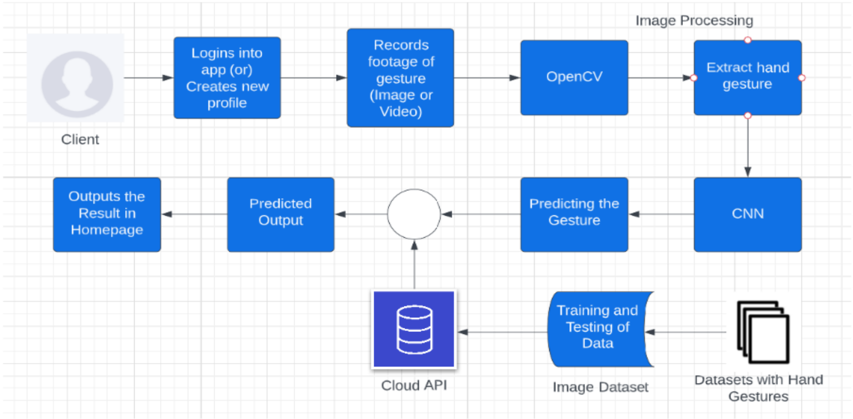
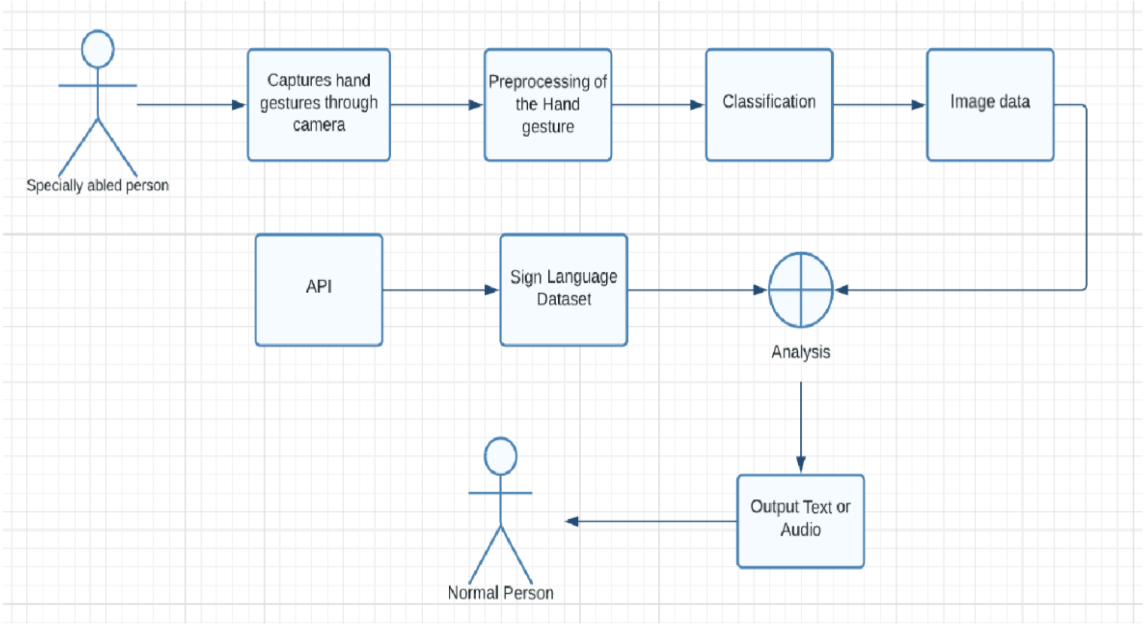


Image Processing
Client
Logins into app (or) Creates new profile
Records footage of gesture (Image or Video)
OpenCV
Extract hand gesture
CNN
Predicting the Gesture
Predicted Output
Outputs the Result in Homepage
Cloud API
Training and Testing of Data
Image Dataset
Datasets with Hand Gestures

5.2) SOLUTION AND TECHNICAL ARCHITECTURE

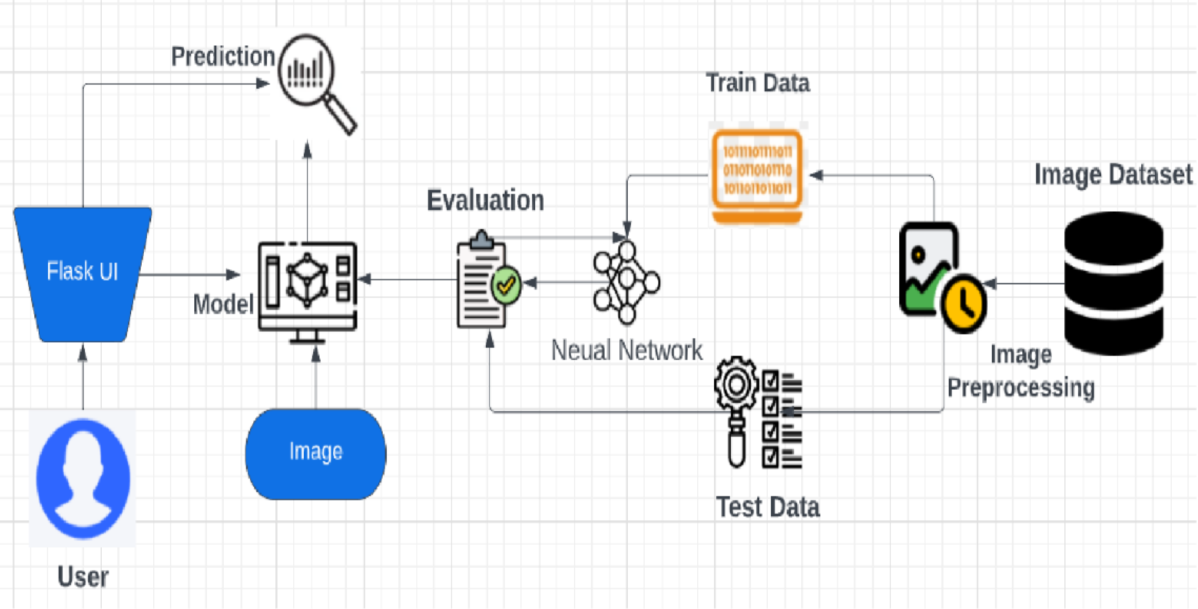
Solution Architecture Diagram:



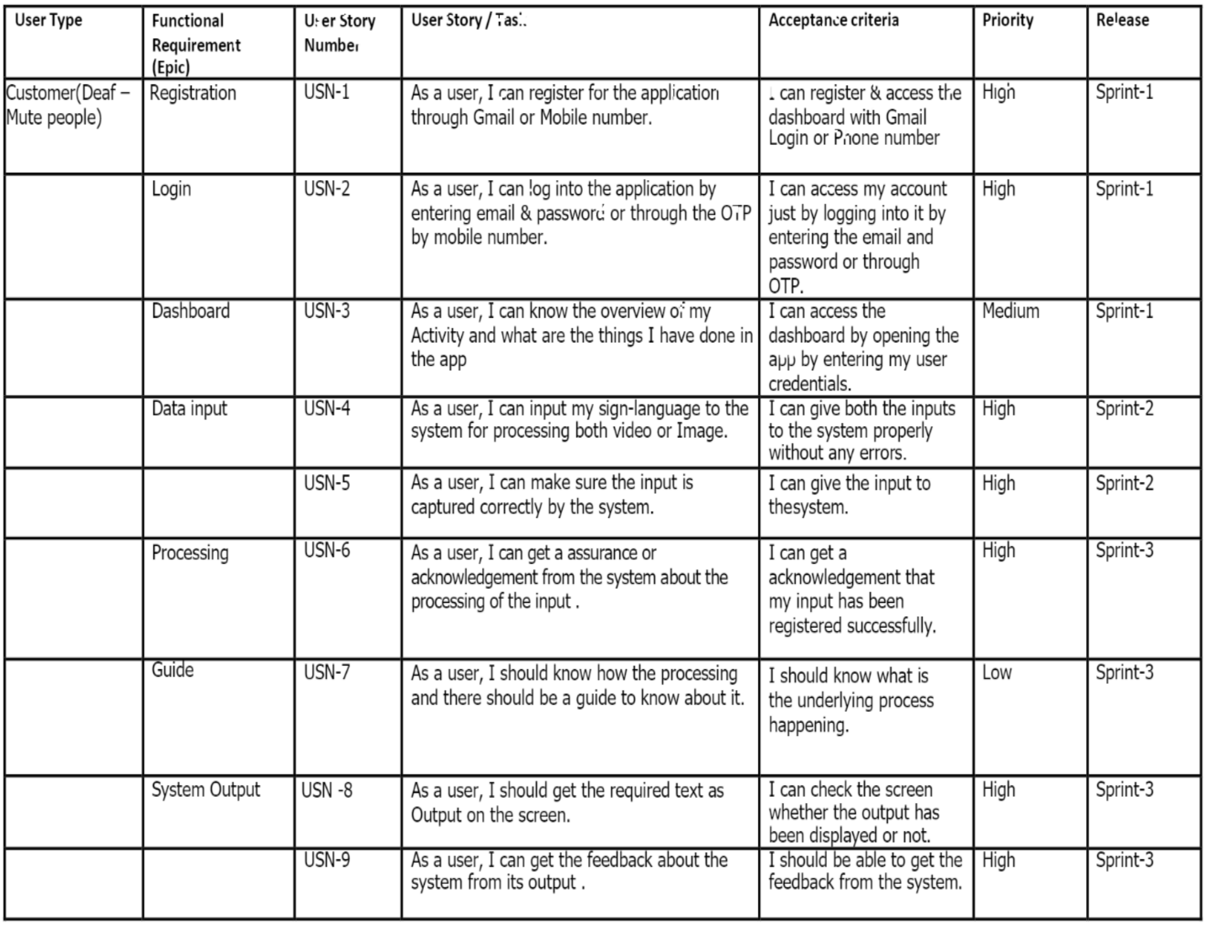
Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Real time communication system powered by AI for specially disabled



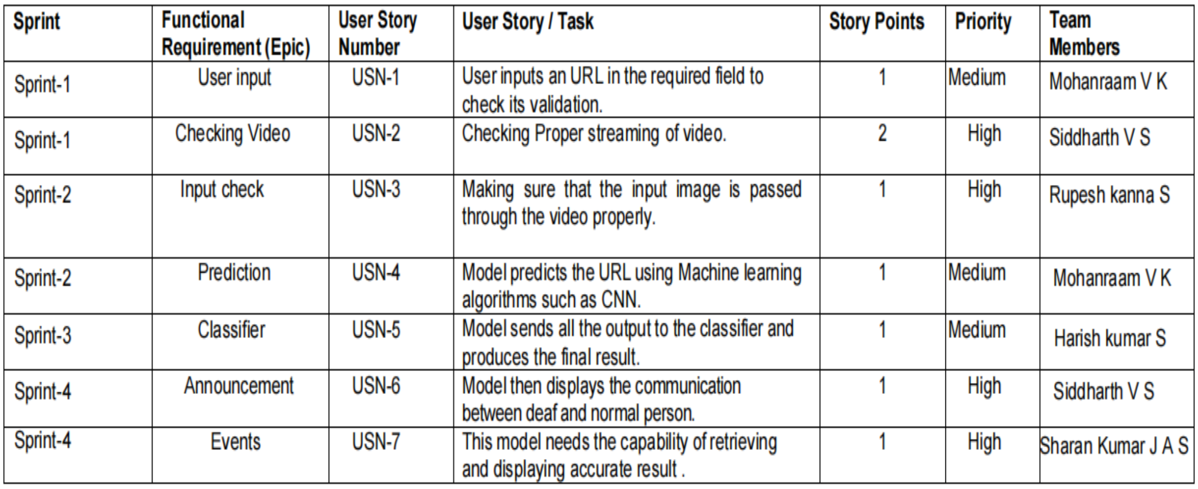
5.3) USER STORIES



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer(Deaf – Mute people)	Registration	USN-1	As a user, I can register for the application through Gmail or Mobile number.	I can register & access the dashboard with Gmail Login or Phone number	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password or through the OTP by mobile number.	I can access my account just by logging into it by entering the email and password or through OTP.	High	Sprint-1
	Dashboard	USN-3	As a user, I can know the overview of my Activity and what are the things I have done in the app	I can access the dashboard by opening the app by entering my user credentials.	Medium	Sprint-1
	Data input	USN-4	As a user, I can input my sign-language to the system for processing both video or Image.	I can give both the inputs to the system properly without any errors.	High	Sprint-2
		USN-5	As a user, I can make sure the input is captured correctly by the system.	I can give the input to thesystem.	High	Sprint-2
	Processing	USN-6	As a user, I can get a assurance or acknowledgement from the system about the processing of the input .	I can get a acknowledgement that my input has been registered successfully.	High	Sprint-3
	Guide	USN-7	As a user, I should know how the processing and there should be a guide to know about it.	I should know what is the underlying process happening.	Low	Sprint-3
	System Output	USN -8	As a user, I should get the required text as Output on the screen.	I can check the screen whether the output has been displayed or not.	High	Sprint-3
		USN-9	As a user, I can get the feedback about the system from its output .	I should be able to get the feedback from the system.	High	Sprint-3

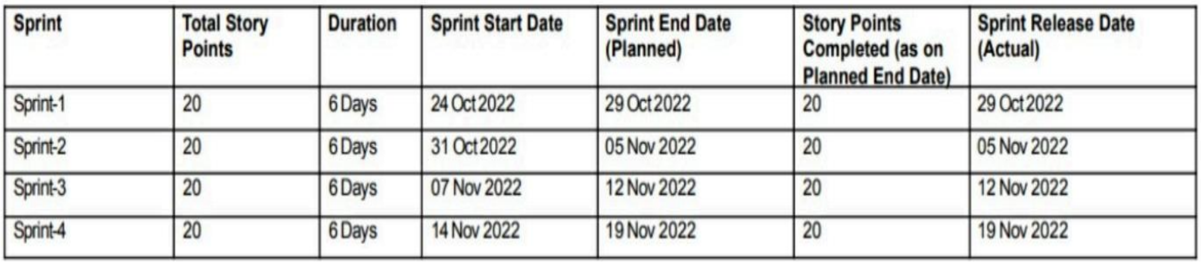
6) PROJECT PLANNING AND SCHEDULING

6.1) SPRINT PLANNING AND ESTIMATION



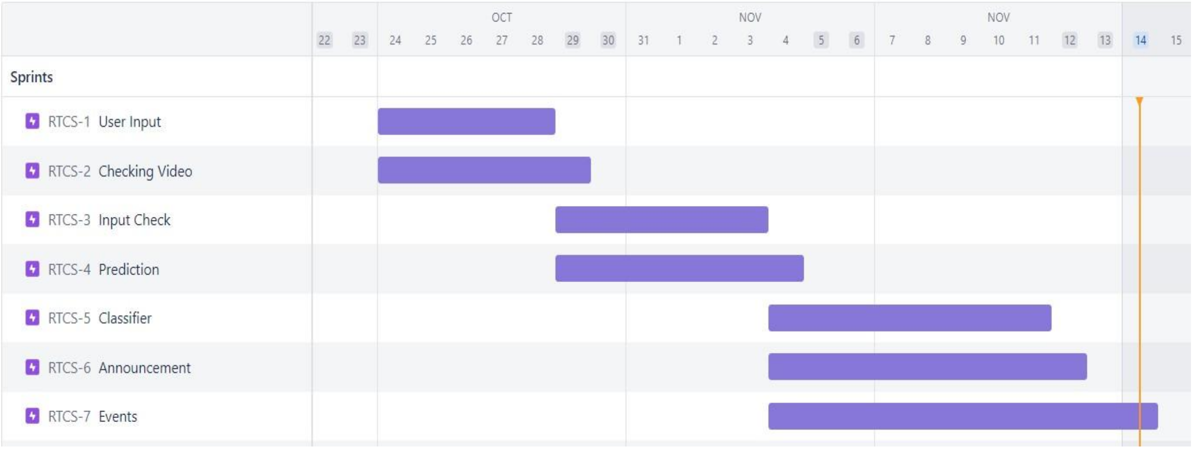
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User input	USN-1	User inputs an URL in the required field to check its validation.	1	Medium	Mohanraam V K
Sprint-1	Checking Video	USN-2	Checking Proper streaming of video.	2	High	Siddharth V S
Sprint-2	Input check	USN-3	Making sure that the input image is passed through the video properly.	1	High	Rupesh kanna S
Sprint-2	Prediction	USN-4	Model predicts the URL using Machine learning algorithms such as CNN.	1	Medium	Mohanraam V K
Sprint-3	Classifier	USN-5	Model sends all the output to the classifier and produces the final result.	1	Medium	Harish kumar S
Sprint-4	Announcement	USN-6	Model then displays the communication between deaf and normal person.	1	High	Siddharth V S
Sprint-4	Events	USN-7	This model needs the capability of retrieving and displaying accurate result .	1	High	Sharan Kumar J A S

6.2) SPRINT DELIVERY SCHEDULE



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3) REPORTS FROM JIRA



Sprints

RTCS-1 User Input

RTCS-2 Checking Video

RTCS-3 Input Check

RTCS-4 Prediction

RTCS-5 Classifier

RTCS-6 Announcement

RTCS-7 Events

7) CODING AND SOLUTIONING

7.1) Feature 1

We added numbers and alphabets hand signs as image inputs using hand signs

The below code is used for this feature

Main code

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
train_data=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True)
test_data=ImageDataGenerator(rescale=1./255,validation_split=0.5)
xtrain=train_data.flow_from_directory('E:/Projects/Jyupter/Dataset/Hand Sign/Train',
target_size=(64,64),class_mode='categorical',batch_size=100)
xtest=test_data.flow_from_directory('E:/Projects/Jyupter/Dataset/Hand Sign/Test',
target_size=(64,64),class_mode='categorical',batch_size=100)

#Layering
model=Sequential()
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(36,activation='softmax'))

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


#Fit
model.fit(xtrain,steps_per_epoch=len(xtrain),
epochs=25,validation_data=xtest,validation_steps=len(xtest))
model.save('Hand-SignV2.h5')

#Test
from tensorflow.keras.preprocessing import image
import numpy as np
fl_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1_9_left_seg_4_cropped.jpeg'
img=image.load_img(fl_img,target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(model.predict(x))
print(pred)
```

```
op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
op[pred]
fl_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test_set/A/15.png'
img=image.load_img(fl_img,target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(model.predict(x))
print(pred)
op=['A','B','C','D','E','F','G','H','I']
op[pred]
import tensorflow.keras.models
new_model = tensorflow.keras.models.load_model('Hand-SignV2.h5')
fl_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test_set/A/15.png'
img=image.load_img(fl_img,target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(new_model.predict(x))
print(pred)
op=['A','B','C','D','E','F','G','H','I']
op[pred]
fl_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1_9_left_seg_4_cropped.jpeg'
img=image.load_img(fl_img,target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(new_model.predict(x))
print(pred)
```

```
op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q',
'R','S','T','U','V','W','X','Y','Z']

op[pred]
```

app.py

```
#Import necessary libraries

from flask import Flask, render_template, Response,url_for

import cv2

from keras.preprocessing import image

from keras.preprocessing.image import load_img

import numpy as np

from keras.models import load_model


#Initialize the Flask app

app = Flask(_name_)

@app.route('/')

def index():

    return render_template('index.html')

@app.route('/video_feed')

def video_feed():

    new_model = load_model('Hand-SignV2.h5')

    cap = cv2.VideoCapture(0)

    try:

        while True:

            cap.set(cv2.CAP_PROP_POS_MSEC,1000*1000)

            ret, frame = cap.read()

            frame = cv2.flip(frame, 1)

            cv2.imshow('Input', frame)

            if ret:

                cv2.imwrite("image.jpeg",frame)

                img = image.load_img("./image.jpeg", target_size=(64, 64))
            x = image.img_to_array(img)


            x = np.expand_dims(x, axis=0)
            pred = np.argmax(new_model.predict(x))
            op = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
          cv2.waitKey(250)
          yield op[pred]
    except KeyboardInterrupt:
        return -1
if _name=='__main_':
    app.run(debug=True)
```

new1.css

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');
body{
   background: #FC7307;
}
a{
   color: blue;
   text-align: center;
   display: block;
   font-size:1.8em;
}
.two{
  margin-left: auto;
  margin-right: auto;
}
```

```
h1{
color : #236AB9;
}
h2{
  color : #341C09;
}
```

7.2) Feature 2

We used our webcam live feed as input. So when the user shows a hand sign the CNN predicts the correct alphabet or number and prints as a text

The below code is used for this feature

feed.html

```
<html>
   <body style="color:#D4E4F7 ">
   {{ cv2.capture(0) }}
   <br>
   {% for i in video_feed() %}
       <h4>{{ i }}</h4>
   {% endfor %}
   </body>
</html>
```

index.html

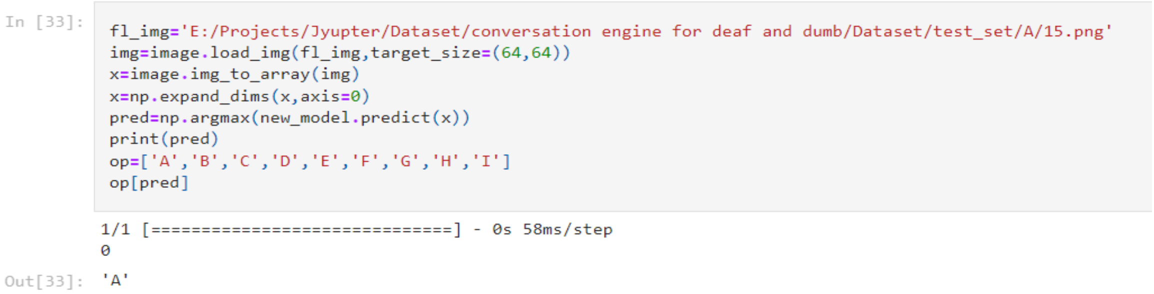
```
<html>
   <head>
       <link rel= "stylesheet" type= "text/css" href= "{{
url_for('static',filename='new1.css') }}">


       Video Streaming Live Web Cam
   </head>
```

```
    <title>Real-Time Communication System Powered by AI for Specially
Abled</title>

    <style>

            h1 {text-align: center;}

      h2 {text-align: center;}

      a {text-align: center;}

    </style>

    <body>

        <div class="container">

            <div class="row">

                <h1 style="font-size:45px">Real-Time Communication System Powered by
AI for Specially Abled</h1>

                <h2 style="font-size:35px">Live Streaming</h3>

                <a href="{{ url_for('video_feed') }}" class="btn btn-warning">Start</a>

            </div>

        </div>

        <img class="one" src="{{ url_for('static', filename='hand-sign.jpg') }}"
width="1000" height="400">

    </body>

</html>
```

8.) TESTING

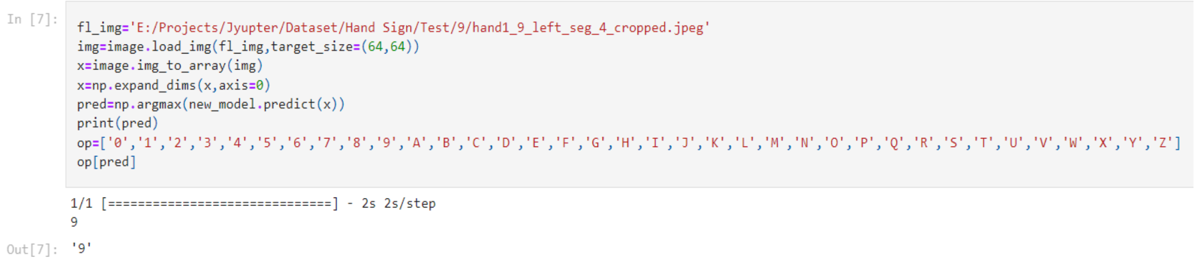
8.1)TEST CASES

TEST CASE 1



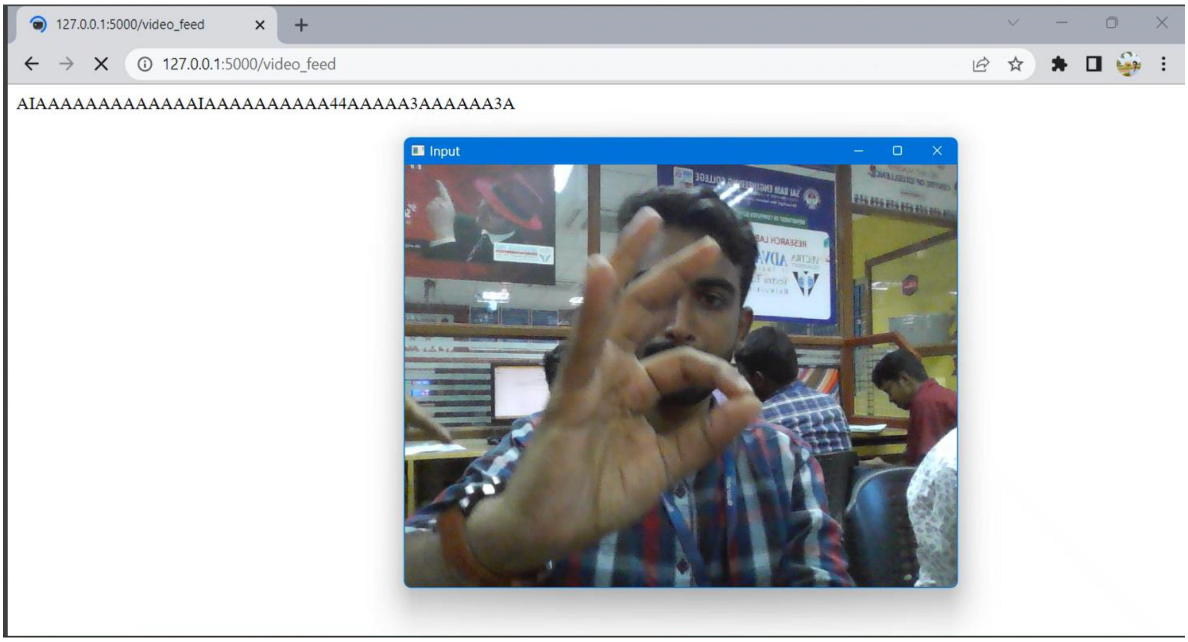
```
In [33]:
fl_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test_set/A/15.png'
img=image.load_img(fl_img,target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(new_model.predict(x))
print(pred)
op=['A','B','C','D','E','F','G','H','I']
op[pred]

1/1 [==============================] - 0s 58ms/step
0
Out[33]: 'A'
```

TEST CASE 2



```
In [7]:
fl_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1_9_left_seg_4_cropped.jpeg'
img=image.load_img(fl_img,target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(new_model.predict(x))
print(pred)
op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
op[pred]

1/1 [==============================] - 2s 2s/step
9
Out[7]: '9'
```

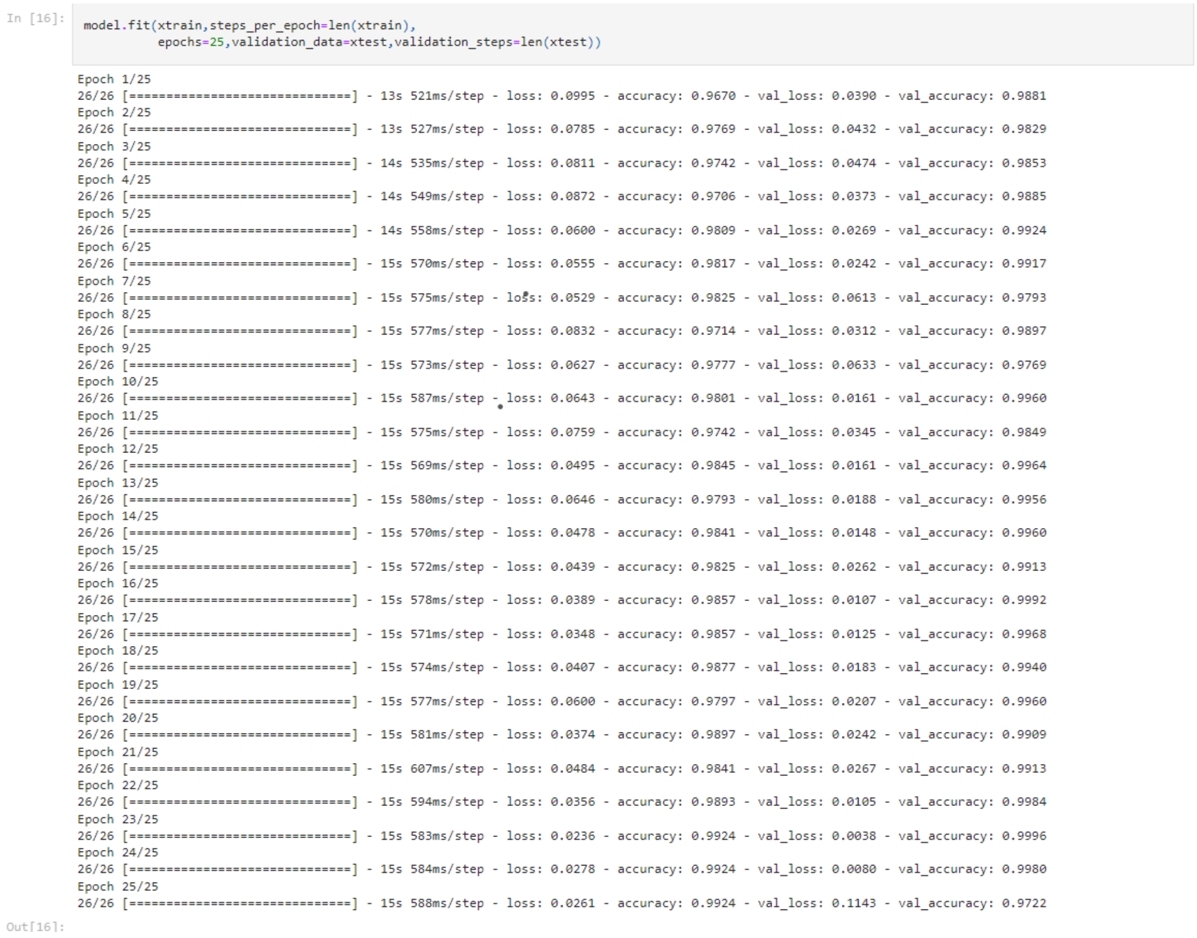
8.2) USER ACCEPTANCE TESTING



AIAAAAAAAAAAAAAIAAAAAAAAAA44AAAAA3AAAAAA3A

9) RESULTS

9.1) PERFORMANCE METRICS



```
In [16]:  model.fit(xtrain,steps_per_epoch=len(xtrain),
                    epochs=25,validation_data=xtest,validation_steps=len(xtest))

          Epoch 1/25
          26/26 [==============================] - 13s 521ms/step - loss: 0.0995 - accuracy: 0.9670 - val_loss: 0.0390 - val_accuracy: 0.9881
          Epoch 2/25
          26/26 [==============================] - 13s 527ms/step - loss: 0.0785 - accuracy: 0.9769 - val_loss: 0.0432 - val_accuracy: 0.9829
          Epoch 3/25
          26/26 [==============================] - 14s 535ms/step - loss: 0.0811 - accuracy: 0.9742 - val_loss: 0.0474 - val_accuracy: 0.9853
          Epoch 4/25
          26/26 [==============================] - 14s 549ms/step - loss: 0.0872 - accuracy: 0.9706 - val_loss: 0.0373 - val_accuracy: 0.9885
          Epoch 5/25
          26/26 [==============================] - 14s 558ms/step - loss: 0.0600 - accuracy: 0.9809 - val_loss: 0.0269 - val_accuracy: 0.9924
          Epoch 6/25
          26/26 [==============================] - 15s 570ms/step - loss: 0.0555 - accuracy: 0.9817 - val_loss: 0.0242 - val_accuracy: 0.9917
          Epoch 7/25
          26/26 [==============================] - 15s 575ms/step - loss: 0.0529 - accuracy: 0.9825 - val_loss: 0.0613 - val_accuracy: 0.9793
          Epoch 8/25
          26/26 [==============================] - 15s 577ms/step - loss: 0.0832 - accuracy: 0.9714 - val_loss: 0.0312 - val_accuracy: 0.9897
          Epoch 9/25
          26/26 [==============================] - 15s 573ms/step - loss: 0.0627 - accuracy: 0.9777 - val_loss: 0.0633 - val_accuracy: 0.9769
          Epoch 10/25
          26/26 [==============================] - 15s 587ms/step - loss: 0.0643 - accuracy: 0.9801 - val_loss: 0.0161 - val_accuracy: 0.9960
          Epoch 11/25
          26/26 [==============================] - 15s 575ms/step - loss: 0.0759 - accuracy: 0.9742 - val_loss: 0.0345 - val_accuracy: 0.9849
          Epoch 12/25
          26/26 [==============================] - 15s 569ms/step - loss: 0.0495 - accuracy: 0.9845 - val_loss: 0.0161 - val_accuracy: 0.9964
          Epoch 13/25
          26/26 [==============================] - 15s 580ms/step - loss: 0.0646 - accuracy: 0.9793 - val_loss: 0.0188 - val_accuracy: 0.9956
          Epoch 14/25
          26/26 [==============================] - 15s 570ms/step - loss: 0.0478 - accuracy: 0.9841 - val_loss: 0.0148 - val_accuracy: 0.9960
          Epoch 15/25
          26/26 [==============================] - 15s 572ms/step - loss: 0.0439 - accuracy: 0.9825 - val_loss: 0.0262 - val_accuracy: 0.9913
          Epoch 16/25
          26/26 [==============================] - 15s 578ms/step - loss: 0.0389 - accuracy: 0.9857 - val_loss: 0.0107 - val_accuracy: 0.9992
          Epoch 17/25
          26/26 [==============================] - 15s 571ms/step - loss: 0.0348 - accuracy: 0.9857 - val_loss: 0.0125 - val_accuracy: 0.9968
          Epoch 18/25
          26/26 [==============================] - 15s 574ms/step - loss: 0.0407 - accuracy: 0.9877 - val_loss: 0.0183 - val_accuracy: 0.9940
          Epoch 19/25
          26/26 [==============================] - 15s 577ms/step - loss: 0.0600 - accuracy: 0.9797 - val_loss: 0.0207 - val_accuracy: 0.9960
          Epoch 20/25
          26/26 [==============================] - 15s 581ms/step - loss: 0.0374 - accuracy: 0.9897 - val_loss: 0.0242 - val_accuracy: 0.9909
          Epoch 21/25
          26/26 [==============================] - 15s 607ms/step - loss: 0.0484 - accuracy: 0.9841 - val_loss: 0.0267 - val_accuracy: 0.9913
          Epoch 22/25
          26/26 [==============================] - 15s 594ms/step - loss: 0.0356 - accuracy: 0.9893 - val_loss: 0.0105 - val_accuracy: 0.9984
          Epoch 23/25
          26/26 [==============================] - 15s 583ms/step - loss: 0.0236 - accuracy: 0.9924 - val_loss: 0.0038 - val_accuracy: 0.9996
          Epoch 24/25
          26/26 [==============================] - 15s 584ms/step - loss: 0.0278 - accuracy: 0.9924 - val_loss: 0.0080 - val_accuracy: 0.9980
          Epoch 25/25
          26/26 [==============================] - 15s 588ms/step - loss: 0.0261 - accuracy: 0.9924 - val_loss: 0.1143 - val_accuracy: 0.9722
Out[16]:
```

10.) ADVANTAGES AND DISADVANTAGES

10.1) ADVANTAGES

*It is very useful for the deaf and dumb(specially abled) person to communicate with other specially abled person or a normal person using this model.

*The acuuracy is higher compared to previous models.

*Before starting the video feed we provide an image which shows what hand sign corresponds to what alphabet , In other words our model is user friendly

10.2) DISADVANTAGES

*Our model is not 100% effective,some flaws in detecting the hand sign may occur.Its mainly because each person use different angles while using hand signs so in detecting the hand sign based on a specific set of data sets is not sufficient.

11.)CONCLUSION

The project has developed 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.

12.)FUTURE SCOPE

Instead of using only hand signs to communicate. We can introduce basic waving motions of hands , for example waving our hand in horizontal motion means hi etc,
It leads to faster and better way of communication.
Instead of using CNN there can be a chance for a new algorithm in the future which has more accuracy.
The scope for this model in the future is high since there are not many applications for the specially abled persons(dead and dumb) to communicate with one another or with a normal person.

13.)APPENDIX

SOURCE CODE

Main code

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

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense

train_data=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True)

test_data=ImageDataGenerator(rescale=1./255,validation_split=0.5)

xtrain=train_data.flow_from_directory('E:/Projects/Jyupter/Dataset/Hand Sign/Train',

target_size=(64,64),class_mode='categorical',batch_size=100)

xtest=test_data.flow_from_directory('E:/Projects/Jyupter/Dataset/Hand Sign/Test',


target_size=(64,64),class_mode='categorical',batch_size=100)
```

```
#Layering
model=Sequential()
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(36,activation='softmax'))

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


#Fit
model.fit(xtrain,steps_per_epoch=len(xtrain),
epochs=25,validation_data=xtest,validation_steps=len(xtest))
model.save('Hand-SignV2.h5')

#Test
from tensorflow.keras.preprocessing import image
import numpy as np
fl_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1_9_left_seg_4_cropped.jpeg'
img=image.load_img(fl_img,target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(model.predict(x))
print(pred)
```

```
op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

op[pred]

fl_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test_set/A/15.png'

img=image.load_img(fl_img,target_size=(64,64))

x=image.img_to_array(img)

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

pred=np.argmax(model.predict(x))

print(pred)

op=['A','B','C','D','E','F','G','H','I']

op[pred]

import tensorflow.keras.models

new_model = tensorflow.keras.models.load_model('Hand-SignV2.h5')

fl_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test_set/A/15.png'

img=image.load_img(fl_img,target_size=(64,64))

x=image.img_to_array(img)

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

pred=np.argmax(new_model.predict(x))

print(pred)

op=['A','B','C','D','E','F','G','H','I']

op[pred]

fl_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1_9_left_seg_4_cropped.jpeg'

img=image.load_img(fl_img,target_size=(64,64))

x=image.img_to_array(img)

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

pred=np.argmax(new_model.predict(x))

print(pred)

op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
```

```
op[pred]
```

app.py

```
#Import necessary libraries
from flask import Flask, render_template, Response,url_for
import cv2
from keras.preprocessing import image
from keras.preprocessing.image import load_img
import numpy as np
from keras.models import load_model

#Initialize the Flask app
app = Flask(_name_)
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/video_feed')
def video_feed():
    new_model = load_model('Hand-SignV2.h5')
    cap = cv2.VideoCapture(0)
    try:
        while True:
            cap.set(cv2.CAP_PROP_POS_MSEC,1000*1000)
            ret, frame = cap.read()
            frame = cv2.flip(frame, 1)
            cv2.imshow('Input', frame)
            if ret:
                cv2.imwrite("image.jpeg",frame)
                img = image.load_img("./image.jpeg", target_size=(64, 64))
                x = image.img_to_array(img)
                x = np.expand_dims(x, axis=0)
                pred = np.argmax(new_model.predict(x))
                op = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
            cv2.waitKey(250)
            yield op[pred]
    except KeyboardInterrupt:
        return -1
if _name=='__main_':
    app.run(debug=True)
```

new1.css

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');
body{
   background: #FC7307;
}
a{
   color: blue;
   text-align: center;
   display: block;
   font-size:1.8em;
}
.two{
  margin-left: auto;
  margin-right: auto;
}

h1{
```

```
color : #236AB9;

}

h2{

  color : #341C09;

}
```

feed.html

```
<html>

   <body style="color:#D4E4F7 ">

   {{ cv2.capture(0) }}

   <br>

   {% for i in video_feed() %}

      <h4>{{ i }}</h4>

   {% endfor %}

   </body>

</html>
```

index.html

```
<html>

   <head>

      <link rel= "stylesheet" type= "text/css" href= "{{ url_for('static',filename='new1.css') }}">


      Video Streaming Live Web Cam

   </head>


   <title>Real-Time Communication System Powered by AI for Specially Abled</title>

   <style>

         h1 {text-align: center;}

    h2 {text-align: center;}
   a {text-align: center;}
  </style>
  <body>
    <div class="container">
      <div class="row">
        <h1 style="font-size:45px">Real-Time Communication System Powered by
AI for Specially Abled</h1>
        <h2 style="font-size:35px">Live Streaming</h3>
        <a href="{{ url_for('video_feed') }}" class="btn btn-warning">Start</a>
      </div>
    </div>
    <img class="one" src="{{ url_for('static', filename='hand-sign.jpg') }}"
width="1000" height="400">
  </body>
</html>
```

Project_Default.xml

```
<component name="InspectionProjectProfileManager">
 <profile version="1.0">
  <option name="myName" value="Project Default" />
  <inspection_tool class="PyPep8Inspection" enabled="true" level="WEAK
WARNING" enabled_by_default="true">
   <option name="ignoredErrors">
    <list>
     <option value="E302" />
     <option value="E305" />
    </list>
   </option>
  </inspection_tool>
  <inspection_tool class="PyPep8NamingInspection" enabled="true" level="WEAK
WARNING" enabled_by_default="true">
   <option name="ignoredErrors">
    <list>
     <option value="N801" />
    </list>
   </option>
  </inspection_tool>
 </profile>
```

```
</component>
```

profiles_settings.xml

```
<component name="InspectionProjectProfileManager">
  <settings>
    <option name="USE_PROJECT_PROFILE" value="false" />
    <version value="1.0" />
  </settings>
</component>
```

IBM.iml

```
<?xml version="1.0" encoding="UTF-8"?>
<module type="PYTHON_MODULE" version="4">
  <component name="NewModuleRootManager">
    <content url="file://$MODULE_DIR$" />
    <orderEntry type="jdk" jdkName="Python 3.9" jdkType="Python SDK" />
    <orderEntry type="sourceFolder" forTests="false" />
  </component>
</module>
```

misc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="ProjectRootManager" version="2" project-jdk-name="Python 3.9" project-jdk-type="Python SDK" />
  <component name="PyCharmProfessionalAdvertiser">
    <option name="shown" value="true" />
  </component>
</project>
```

modules.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="ProjectModuleManager">
    <modules>
      <module fileurl=file://$PROJECT_DIR$/.idea/IBM.iml
filepath="$PROJECT_DIR$/.idea/IBM.iml" />
    </modules>
  </component>
</project>
```

GITHUB LINK:- https://github.com/IBM-EPBL/IBM-Project-24768-1659948659

DEMO LINK:- https://www.youtube.com/watch?v=KwFK2SQf3qM