REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

CHAPTER 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. Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and image is given as output.

1.2 PURPOSE

People get to know one another by sharing their ideas, thoughts, and experiences with those around them. There are numerous ways to accomplish this, one of which is the gift of images. Everyone can very convincingly transfer their thoughts and understand each other through images. 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. 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.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

Some of the existing solutions for solving this problem are:

[1] Face Based Real Time Communication for Physically and Speech Disabled People

An improved real-time communication system using machine learning and computer vision. The aim is to create a communication channel between the specially a bled and the society, so they can express there feelings, thoughts and understand other people's feelings and thoughts through real time communication and facial expressions.

[2] Artificial Intelligence and Accessibility

Seeing AI, visually impaired people can easily read their mail by placing documents under the smartphone camera. AI technology can apply to any type of disability profile. For instance, people with reduced mobility can control everything at home.

[3] Survey on application of Artificial Intelligence in Cyber Security

Cyber security refers to protecting your personal computer from malicious software. Machine learning has a lot many algorithms and system which protect users from threats. Such as the Pay pal app which was developed in December 1998, uses machine learning algorithms to protect its users from different threats and online spoofing. It uses three types of machine learning algorithms that are linear, neural network and deep learning algorithm.

[4] Machine Learning based techniques in data analysis

It is an application from which we can virtually explore streets of cities. It uses a dense geo sampling tool to shows the streets of cities. Streets are captured through a fleet of vehicles equipped with a specialized camera. After collection of photos, they are digitally processed and combined together and looks like a single image. From files reported for privacy, Google pixelated faces of pedestrian and license plate which is captured. Web mapping technologies have been embraced by discipline such as geography, archaeology and ecology, but also by several social scientific disciplines. Researchers working in the discipline of geography, archaeology, and ecology quickly incorporated web based mapping technologies into their research designs. There are various applications of google street view in research field, although the number still remains limited. It is also used for better estimation of fish catching, estimation of forestry biomass in India, estimation of area of different regions or lakes, etc.

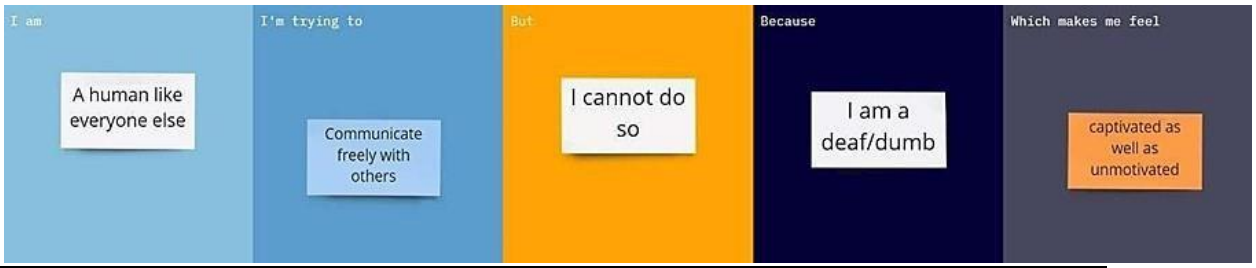
2.2 References

1. Ann, O. C., Lu, M. V., & Thing, L. B. (2011). A face based real time communication for physically and speech disabled people. In Assistive and Augmentative Communication for the Disabled: Intelligent Technologies for Communication, Learning and Teaching (pp. 70-102). IGI Global.
2. Azmi, A., Alsabhan, N. M., & AlDosari, M. S. (2009). The Wiimote with SAPI: Creating an accessible low-cost, human computer interface for the physically disabled. International Journal of Computer Science and Network Security, 9(12), 63-68.
3. Li, J. H. (2018). Cyber security meets artificial intelligence: a survey. Frontiers of Information Technology & Electronic Engineering, 19(12), 1462-1474.
4. Machine Learning based techniques in data analysis (Lavanya Vemulapalli, Dr.P.Chandra Sekhar – 2018)

2.3 Problem Statement Definition

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 images are given as output.

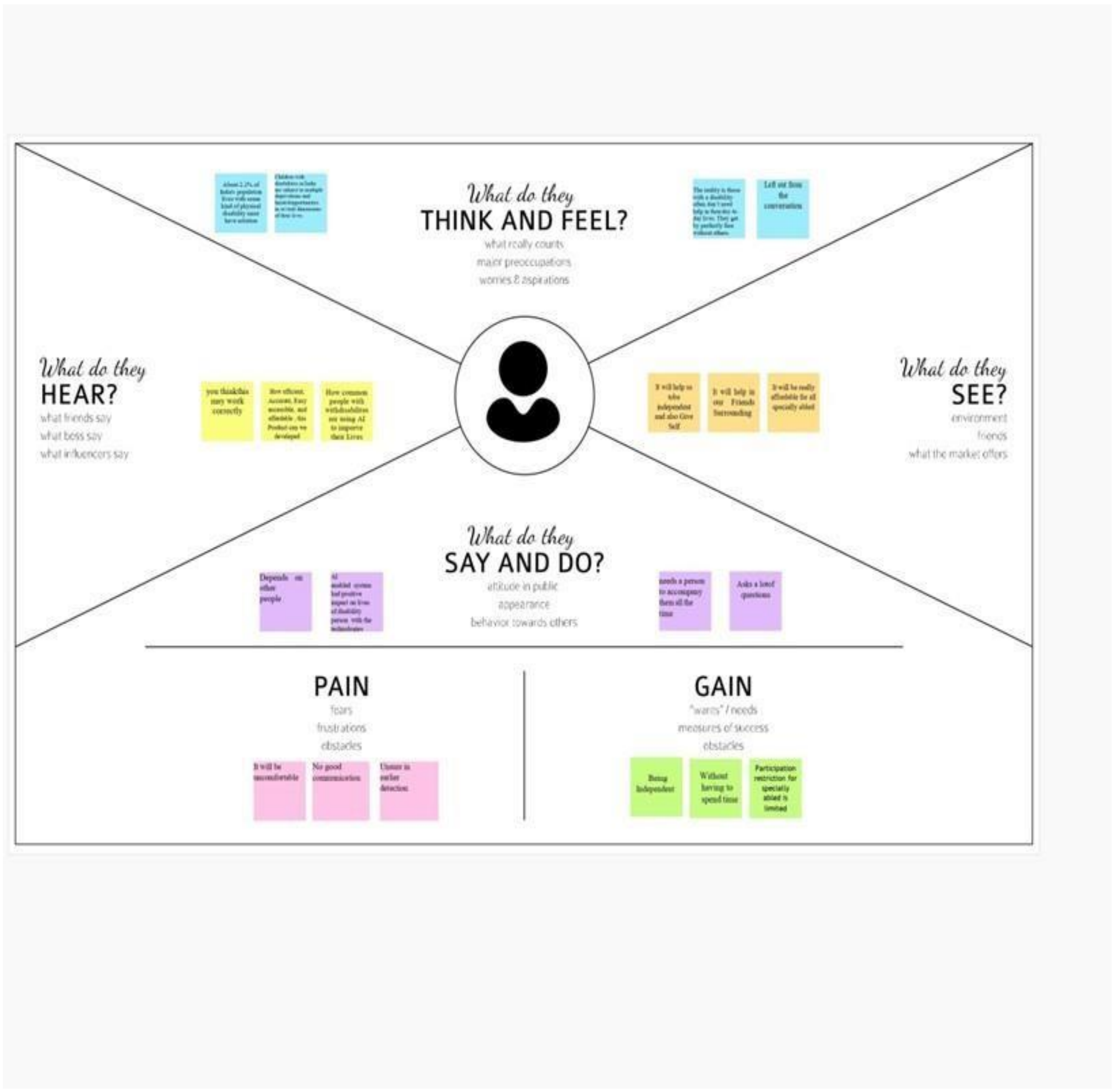
Example:


I am
A human like everyone else
I'm trying to
Communicate freely with others
But
I cannot do so
Because
I am a deaf/dumb
Which makes me feel
captivated as well as unmotivated

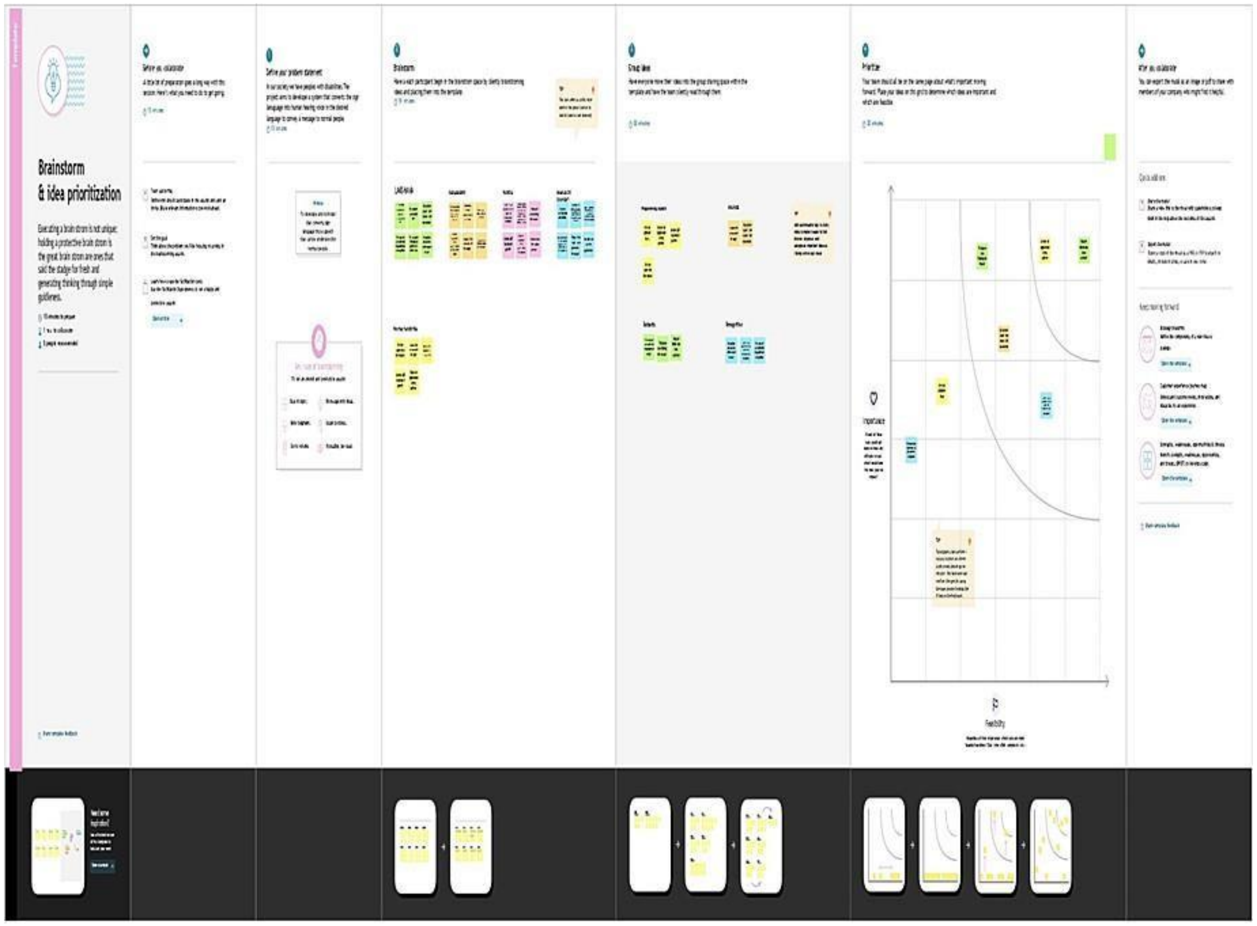
Problem Stateme nt (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Person withoutdisabili ty	Communica te with deaf and dumb via phone	It is not possib le	I can't able to understa nd the sign langua ge	Frustrated
PS-2	A human like everyone	Communicate freely with others	I cannot do so	I am a deaf/du mb	Cap vated as well as unmo vated

CHAPTER 3
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

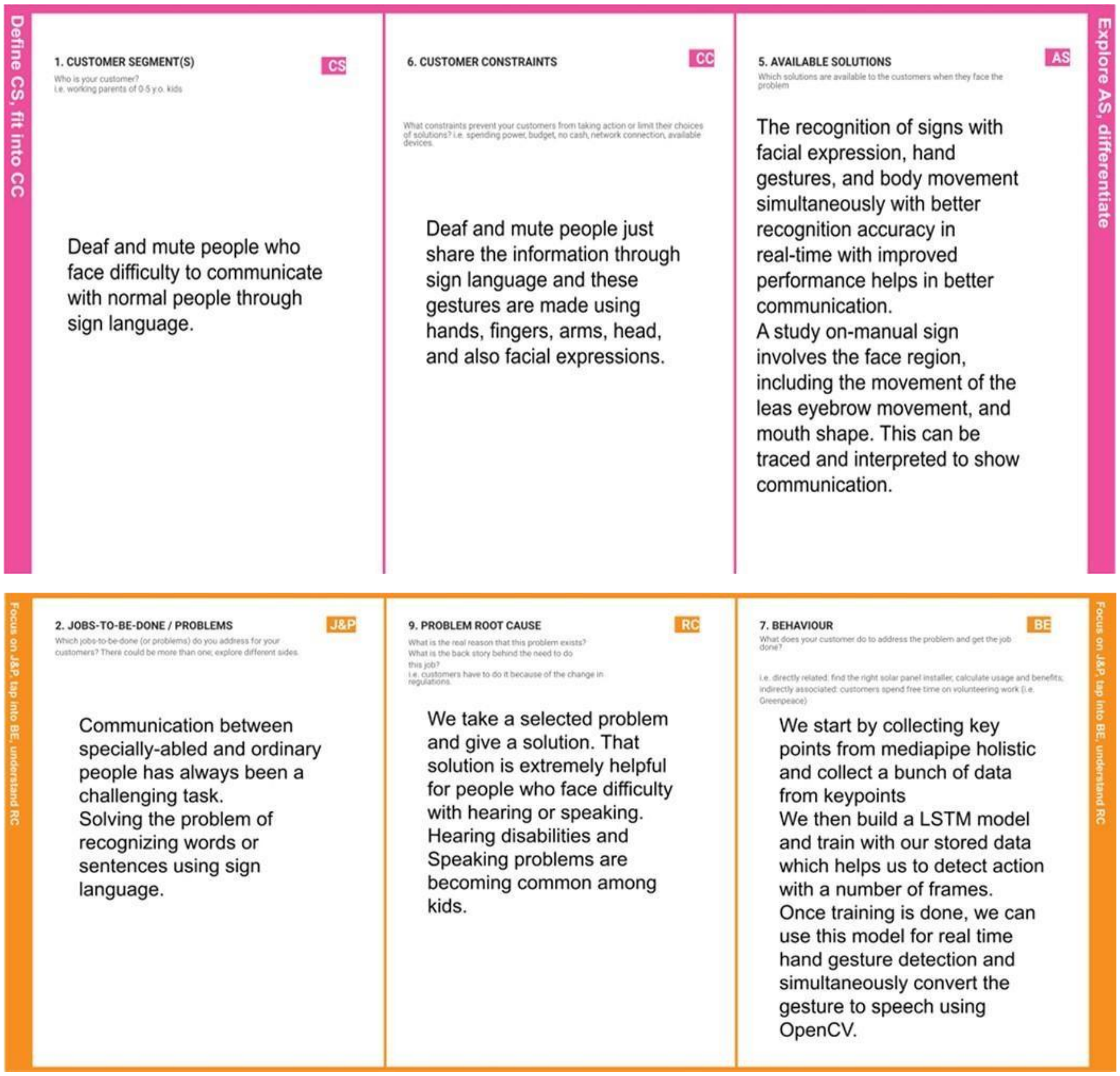


3.3 Proposed Solution

S.No	Parameter	Description

1.	Problem Statement (Problem to be solved)	Sign Language is a communication method used by people with hearing disability or speaking disability using hand gesture. Since it is not know by everyone people with disabilitytents to face difficulty while communicating. The proposed system is to help them convert the gesture-based sign language to voice based message
2.	Idea / Solution description	The idea is to identify the live gestured basedsign language and to translate it into the voice- based message to make the communication easy for the differently abled people
3.	Novelty / Uniqueness	The idea is to create a system that will ease out the processes of conversion of sign language to hearable voice message. The application is trained withevery gesture possible.
4.	Social Impact/ Customer Satisfaction	1. To boost the confidence of a differently abledperson by making them independent 2. To breakthe communication barrier 3. Ease the conversion of sign language to voice-based message 4. To improvethe everyday livesof people withdisabilities
5.	Business Model(Revenue Model)	The targeted customers of this system are people with hearing disability and speaking disability and the people around them Because of its uniqueness and the essentiality undoubtedly the market of the system will be huge
6.	Scalability of the Solution	The proposed application for the people with disability is accessible in desktops, mobile phones aroundthe globe.

3.4 Problem Solution fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Deaf and mute people who face difficulty to communicate with normal people through sign language.

6. CUSTOMER CONSTRAINTS CC

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

Deaf and mute people just share the information through sign language and these gestures are made using hands, fingers, arms, head, and also facial expressions.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem

The recognition of signs with facial expression, hand gestures, and body movement simultaneously with better recognition accuracy in real-time with improved performance helps in better communication.
A study on-manual sign involves the face region, including the movement of the leas eyebrow movement, and mouth shape. This can be traced and interpreted to show communication.

Explore AS, differentiate

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

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

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

Communication between specially-abled and ordinary people has always been a challenging task.
Solving the problem of recognizing words or sentences using sign language.

9. PROBLEM ROOT CAUSE RC

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

We take a selected problem and give a solution. That solution is extremely helpful for people who face difficulty with hearing or speaking.
Hearing disabilities and Speaking problems are becoming common among kids.

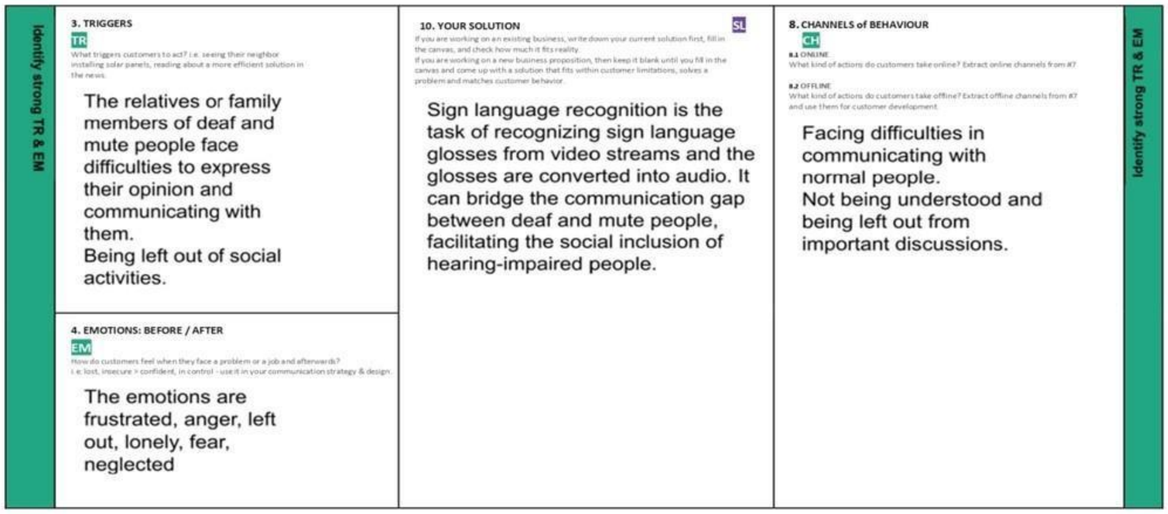
7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

We start by collecting key points from mediapipe holistic and collect a bunch of data from keypoints
We then build a LSTM model and train with our stored data which helps us to detect action with a number of frames.
Once training is done, we can use this model for real time hand gesture detection and simultaneously convert the gesture to speech using OpenCV.

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



Identify strong TR & EM

3. TRIGGERS

TR

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

The relatives or family members of deaf and mute people face difficulties to express their opinion and communicating with them.
Being left out of social activities.

4. EMOTIONS: BEFORE / AFTER

EM

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

The emotions are frustrated, anger, left out, lonely, fear, neglected

10. YOUR SOLUTION

SL

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

Sign language recognition is the task of recognizing sign language glosses from video streams and the glosses are converted into audio. It can bridge the communication gap between deaf and mute people, facilitating the social inclusion of hearing-impaired people.

8. CHANNELS of BEHAVIOUR

CH

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

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Facing difficulties in communicating with normal people.
Not being understood and being left out from important discussions.

Identify strong TR & EM

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirements Hardware Requirements:

Operating System	Windows, Mac, Linux
CPU (for training)	Multi Core Processors (i3 or above/equivalent)
GPU (for training)	NVIDIA AI Capable / Google's TPU
Web Cam	Integrated or External with Full HD Support

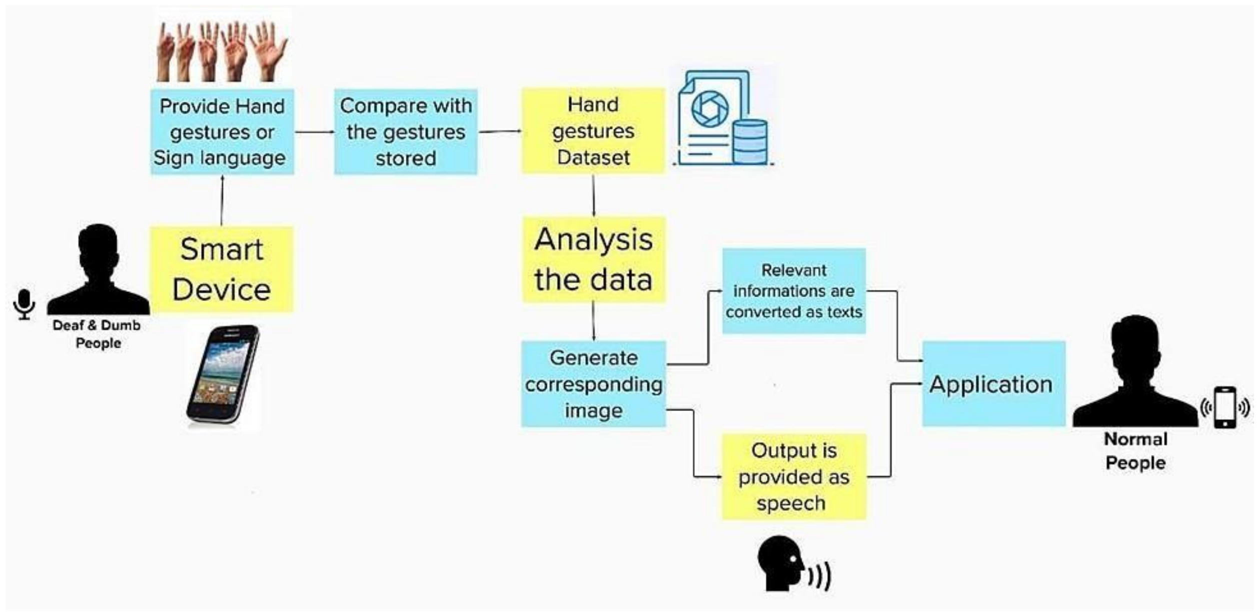
Software Requirements

Python	v3.9.0 or Above
Python Packages	flask, tensorflow, opencv-python, keras, numpy,pandas, virtualenv, pillow
Web Browser	Mozilla Firefox, Google Chrome or any modern web browser
IBM Cloud (for training)	Watson Studio - Model Training & Deployment as Machine Learning Instance

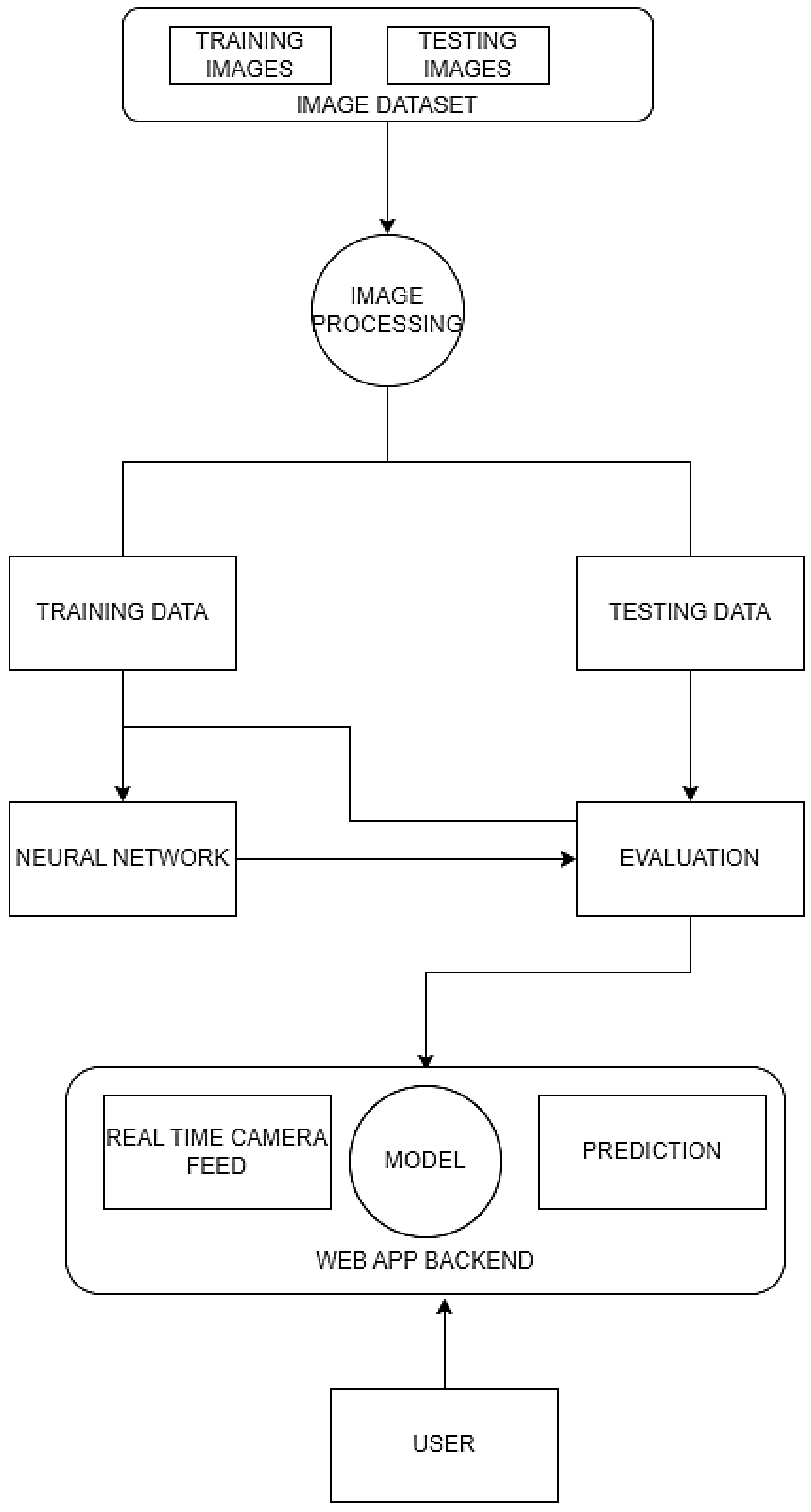
CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagrams


Provide Hand gestures or Sign language
Compare with the gestures stored
Hand gestures Dataset
Smart Device
Deaf & Dumb People
Analysis the data
Relevant informations are converted as texts
Generate corresponding image
Application
Normal People
Output is provided as speech

Data Flow Diagram


TRAINING IMAGES
TESTING IMAGES
IMAGE DATASET
IMAGE PROCESSING
TRAINING DATA
TESTING DATA
NEURAL NETWORK
EVALUATION
REAL TIME CAMERA FEED
MODEL
PREDICTION
WEB APP BACKEND
USER

Flow Chart

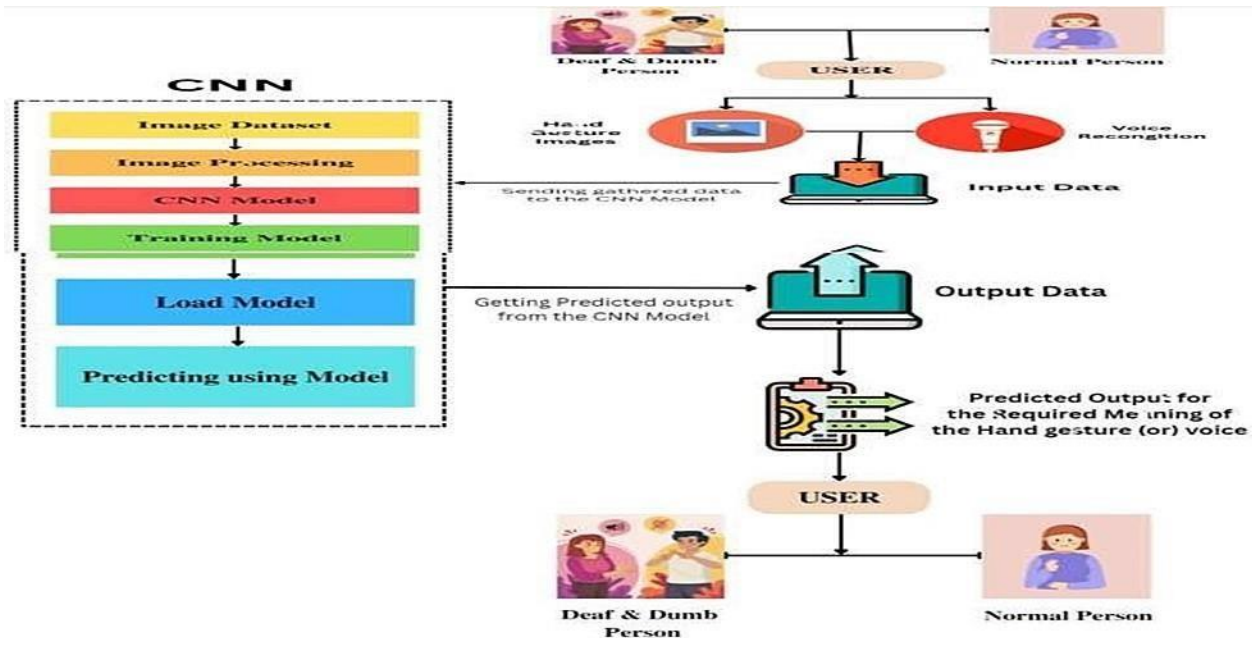
5.2 Solution & Technical Architecture

Solution Architecture:

Solution architecture is a bridge the gap between business problems and technology solutions. Its goals are to:

- The best tech solutionto solve existingbusiness problems.
- Describing the structure, characteristics, behaviour, and other aspects of the softwareto project stakeholders.
- Defined aboutfeatures, development phases,and solution requirements.
- Provided specifications according to which the solution is defined, managed,and delivered.

Example - Solution Architecture Diagram:



5.3 User Stories

User Type	Func onal Requireme nt(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Custom er (Mobile user)	Register with the users information.	USN-1	As a user, I can register for the applica on by entering my email,password, and confirming my password.	I can access my account / dashboard in the applica on.	High	Sprint-1
Custom er (Deaf people)	To communicate with people using signs.	USN-2	As a user, I can see my applica on and made changes in any browser and register to it.	I can login and see my accountin the applica on anywhere at any me.	High	Sprint-1
Customer (Dumb people)	To communicate with people easily and efficiently.	USN-3	As a user, I can see my applica on and made changes in any browsers and register to it.	I can loginand see my account in the applica on anywhere.	High	Sprint-1
Customer (Normal people)	User needs to communicate with specially abled people.	USN-4	As a user, I can register for the applica on by entering my email, password, and confirma on ismade.	I can login and see my account.	Medium	Sprint-2
Customer (Learner of Sign language)	User needs to be aware and learn about sign languag e.	USN-5	As a user, I can create my account in the applica on withmy email and password, to get knowledge about sign languages.	I can create my account andaccess the dashboard in the applica on.	High	Sprint-1
Custom er (Web user)	They want the update on the applica on condi on.	USN-6	As a user, I can register for the applica on by entering my email, password, and confirming my password. To get details about real- me communica on.	I can able to use any browser to access the applica on fromanywhere, to know anything about real- me communica o n.	High	Sprint-1

Custom er Care Executi ve	They want to helppeople by sending applica on condi ons.	USN-7	As a user, I can receivea message from the administra on about condi ons of applica on of real- me communica on.	I will analyse and send SMSto the people.	High	Sprint-1

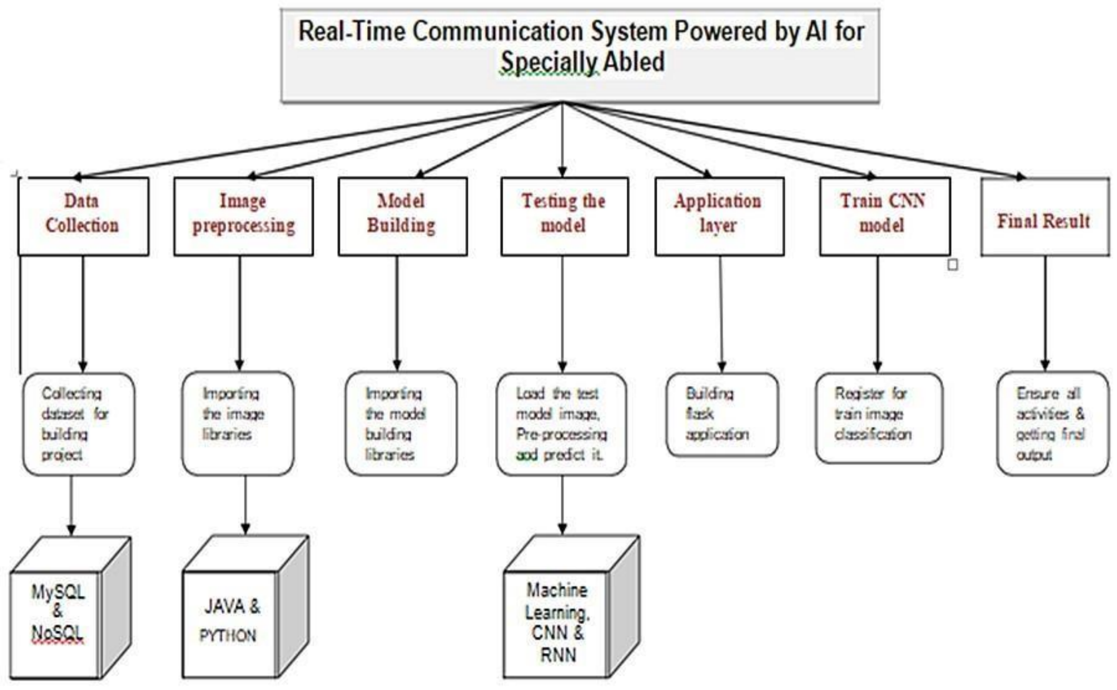
CHAPTER 6 PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning And Estimation

Milestone	**Functional Requirement (Epic)**	**Milestone Story Number**	**Milestone Story/ Task**
Milestone 1	Data Collection	M1	We'recollecting dataset for building our project and creating two folders, onefor training and another one for testing.
Milestone 2	Image Preprocessing	M2	Importing imagedata generator libraries and applying imagedata generator functionality to trainthe test set.
Milestone 3	Building Model	M3	Importing the model building libraries, Initializing the model, AddingConvolution layers, Adding the Pooling layers,Adding the Flattenlayers, Adding Denselayers, Compiling themodel Fit and Save the model.

Milestone 4	Testing Model	M4	Import the packages first.Then we savethe model and Load the test image, preprocess it and predict it.
Milestone 5	Application Layer	M5	Build theflask application andthe HTML pages.
Milestone 6	TrainConversati on Engine	M6	Register forIBM Cloud and train Image Classification Model
Milestone 7	Final Result	M7	To ensureall the activities and resulting the finaloutput.

MILESTONE ACTIVITYPLAN


Real-Time Communication System Powered by AI for Specially Abled
Data Collection
Image preprocessing
Model Building
Testing the model
Application layer
Train CNN model
Final Result
Collecting dataset for building project
Importing the image libraries
Importing the model building libraries
Load the test model image, Pre-processing and predict it.
Building flask application
Register for train image classification
Ensure all activities & getting final output
MySQL & NoSQL
JAVA & PYTHON
Machine Learning, CNN & RNN

SPRINT PLANING

Sprint	Func onal Requirement (Epic)	User Story Number	User Story/Task	Story Points	Priority
Sprint – 1	Dataset Collection	USN – 1	Collect Dataset for building model	9	High
Sprint – 1	Image Preprocessing	USN – 2	Perform preprocessing techniques on thedataset	8	Medium
Sprint – 2	Model Building	USN – 3	Import the required libraries, add the necessary layersand compile the model	10	High
Sprint – 2		USN – 4	Training the image classifica on model using CNN	7	Medium

Sprint – 3	Training and Testing the Model	USN – 5	Training the model and tes ng the model's performance	9	High
Sprint – 4	Applica on Developme nt	USN – 6	Conver ng the input gesture image into English Alphabets	8	Medium

6.2 Sprint Delivery Schedule

Sprint	**Total StoryPoin ts**	**Durati on**	**Sprint StartDate**	**Sprint End Date (Planned)**	**Story Points Complet ed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint – 1	17	6 Days	24 October, 2022	29 October, 2022	17	29 Oct 2022
Sprint – 2	17	6 Days	31 October, 2022	05 November, 2022	17	05 Nov 2022
Sprint – 3	9	6 Days	07 November, 2022	12 November, 2022	9	12 Nov 2022
Sprint – 4	5	6 Days	14 November, 2022	19 November, 2022	8	19 Nov 2022

Velocity

Average Velocity= **Velocity**

Sprint Duration

- Average Velocity → AV
- Velocity → Points per sprint
- Sprint Duration → Number of days per sprint

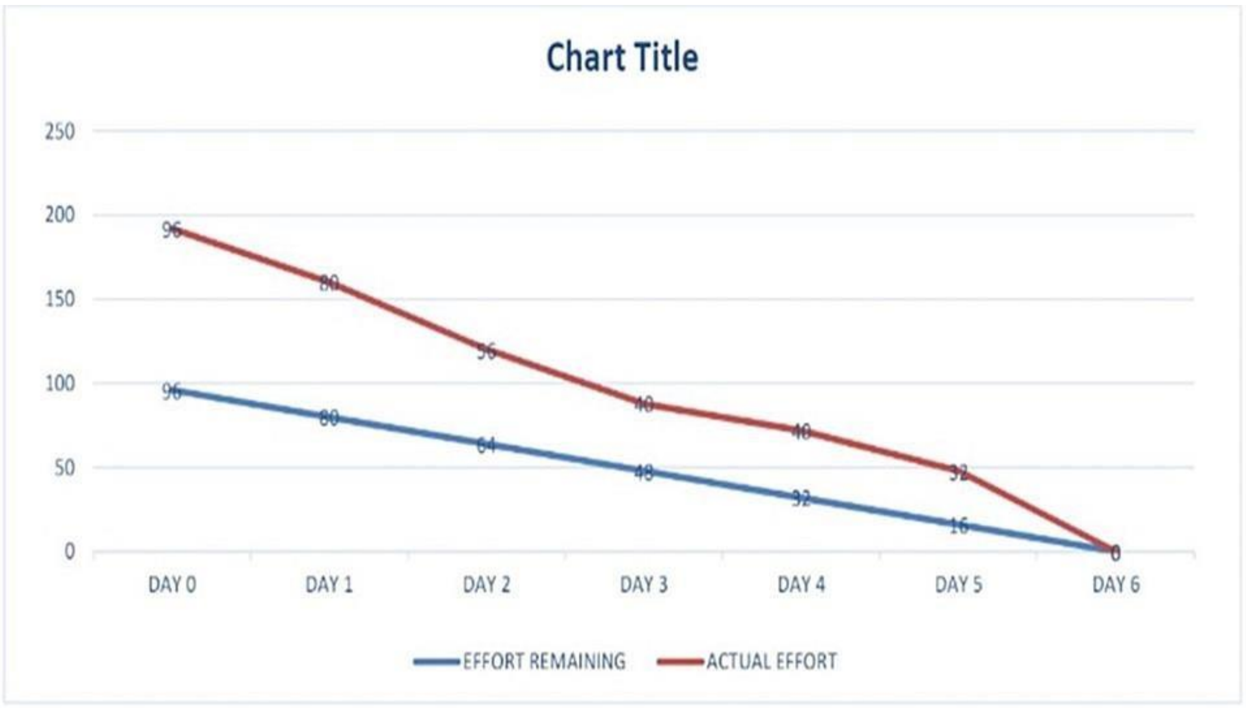
1. Sprint – 1: AV = 17÷6 = 2.83

2. Sprint – 2: AV = 17÷6 = 2.83 '

3. Sprint – 3: AV = 9÷6 = 1.5

4. Sprint – 4: AV = 5÷6 = 0.83

6.3 Report From Jira



BURNDOWN CHART

CHAPTER 7

CODING AND EXECUTION

7.1 Feature 1

The proposed system consists of two features front end and backend. The frontend is designed using HTML and CSS. The first feature is a webpage whenever a user wants to translate the sign language to English, they can go to the webpage it has start button. On pressing the start button, it will turn on the camera for live translation. Once the camera is turned on, we can start translating.

Coding:

```
 <!DOCTYPE html>

 <html>

 <head>

 <title>Real Time Communication</title>

 <style>   body{   background-image: linear-gradient(to bottom
right, blue, black);   background-repeat: no-repeat;
background-attachment: fixed;

 }

 h1,h2,a,p{

color:white;   }
```

```
 </style>

 </head>

 <body>

 <div class="title">

 <h1><center>

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI

FOR SPECIALLY ABLED</center></h1>

</div>

<center><img src="../static/img/img.png" width="300" height="300"></center>  <div>

<center><h2>Show these Gestures to get the Alphabet</h2></center>

</div>

<div>

<center><a href="{{ url_for('predict') }}">CLICK HERE TO SHOW YOUR
GESTURES</a></center>

</div>

<div>

<center>  <p>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.<br>

<br>

The project aims to develop a system that converts the sign language into a alphabet in
the desired language to convey a message to normal people. 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 is
given as output.</p>
</center>
</div>
 </body>
 </html>
```

7.2 Feature 2

The second feature of the proposed system is backend. The backend is designed using python with the packages of python like flask, tensorflow, opencv-python, keras, numpy, pandas, virtualenv, pillow and Machine learning technology and trained with datasets. Once the camera is turned on the system detects and identify the sign language and translate it to English by matching the live action with the trained dataset. **Coding:**

```
from flask import Flask,render_template,request import
cv2  from keras.models import load_model import numpy
as np from gtts import gTTS import os  from
keras.preprocessing import image from skimage.transform
import resize from playsound import playsound app =
Flask(_name_) model=load_model("aslpng1.h5")
vals = ['A', 'B','C','D','E','F','G','H','I']
@app.route('/', methods=['GET'])
def index():
        return
render_template('index.html')
```

```
@app.route('/index', methods=['GET']) def home():

            return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST']) def

predict():

                        print("[INFO] starting video

stream...")                         vs = cv2.VideoCapture(0)

(W, H) = (None, None)                         while True:

            (grabbed, frame) = vs.read()

                                  if not grabbed:

break                                    if W is None or H

is None:

                                              (H, W) = frame.shape[:2]

output = frame.copy()

                                    # r = cv2.selectROI("Slect", output)

                                # print(r)

cv2.rectangle(output, (81, 79), (276,274), (0,255,0), 2)

frame = frame[81:276, 79:274]

                                    frame = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)

                                    _, frame = cv2.threshold(frame, 95, 255,
cv2.THRESH_BINARY_INV)

                                    frame = cv2.cvtColor(frame,

cv2.COLOR_GRAY2RGB)                                    img =
resize(frame,(64,64,3))                              img =

np.expand_dims(img,axis=0)                                    if(np.max(img)>1):

                                          img          =          img/255.0

result                 =                  np.argmax(model.predict(img))

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

result=str(index[result])

                          cv2.putText(output, "The Predicted Letter : {}".format(result), (10,
50), cv2.FONT_HERSHEY_PLAIN,

                                                                 2, (150,0,150), 2)

                                  cv2.putText(output, "Press q to exit", (10,450),
cv2.FONT_HERSHEY_PLAIN, 2, (0,0,255), 2)

                          speech = gTTS(text = result, lang = 'en', slow = False)

cv2.imshow("Output", output)                                   key =

cv2.waitKey(1) & 0xFF                                  if key == ord("q"):

                                          break

print("[INFO] cleaning up...")

vs.release()

cv2.destroyAllWindows()                        return

render_template("index.html") if


_name_ == '_main_':

            app.run(debug=True)
```

CHAPTER 8

TESTING

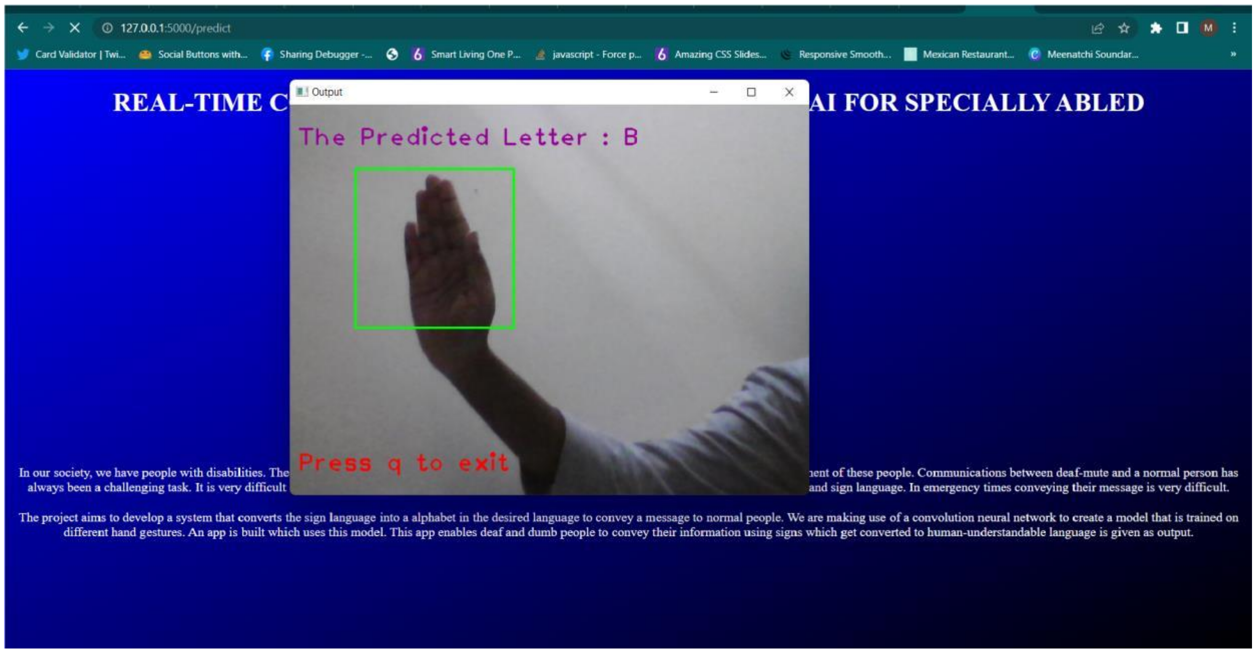
```
# Importing Libarries from
tensorflow.keras.models import load_model from
tensorflow.keras.preprocessing import image
import numpy as np import cv2 # loading model
model = load_model('aslpng1.h5') from
skimage.transform import resize def
detect(frame):
   img = resize(frame, (64, 64, 3))
img = np.expand_dims(img, axis = 0)
if np.max(img) > 1:        img =
img/255.0     prediction =
model.predict(img)
print(prediction)     return prediction
frame = cv2.imread(r"D:\Real-time
Communication System for specially
abled\Dataset\test_set\A\16.png")
data = detect(frame)
```

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

index[np.argmax(data)] # Importing Libraries

import cv2  import numpy as np  from

tensorflow.keras.models import load_model from

tensorflow.keras.preprocessing import image

# Loading Model  model =

load_model("aslpng1.h5") video

= cv2.VideoCapture(0)

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

True:

    success, frame = video.read()      cv2.imwrite('frame.jpg',

frame)      img = image.load_img('frame.jpg', target_size =

(64, 64))      x = image.img_to_array(img)      x =

cv2.cvtColor(x, cv2.COLOR_BGR2HSV)      a =

x.array_to_img(x)      cv2.imshow("")  x =

np.expand_dims(x, axis = 0)


    pred = np.argmax(model.predict(x), axis = 1)  y = pred[0]

copy = frame.copy()     cv2.rectangle(copy, (320, 100), (620,

400), (255, 0, 0), 5)

    cv2.putText(frame, "The Predicted Alphabet : " + str(index[y]), (100, 100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)

cv2.imshow('frame',    frame)                    if

cv2.waitKey(1) & 0xFF == ord('q'):         break

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

CHAPTER 9

RESULT

9.1 Performance Metrics



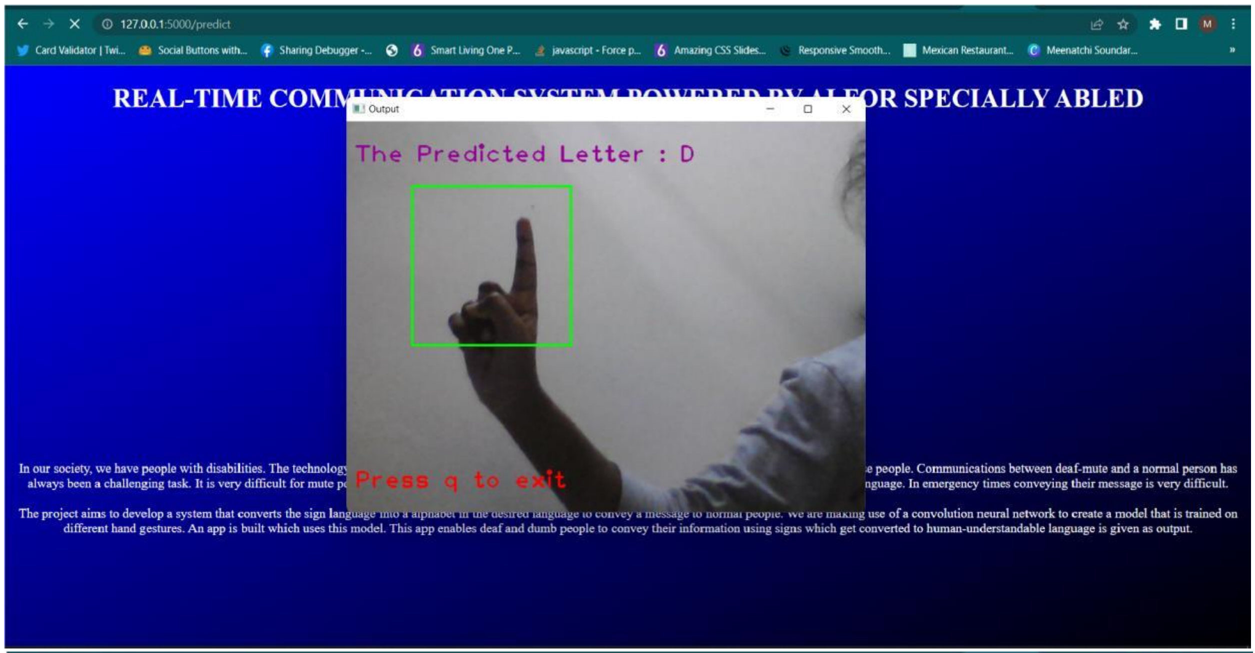
REAL-TIME C AI FOR SPECIALLY ABLED

```
The Predicted Letter : B
```

```
Press q to exit
```

In our society, we have people with disabilities. The ent of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult and sign language. In emergency times conveying their message is very difficult.

The project aims to develop a system that converts the sign language into a alphabet in the desired language to convey a message to normal people. 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 is given as output.



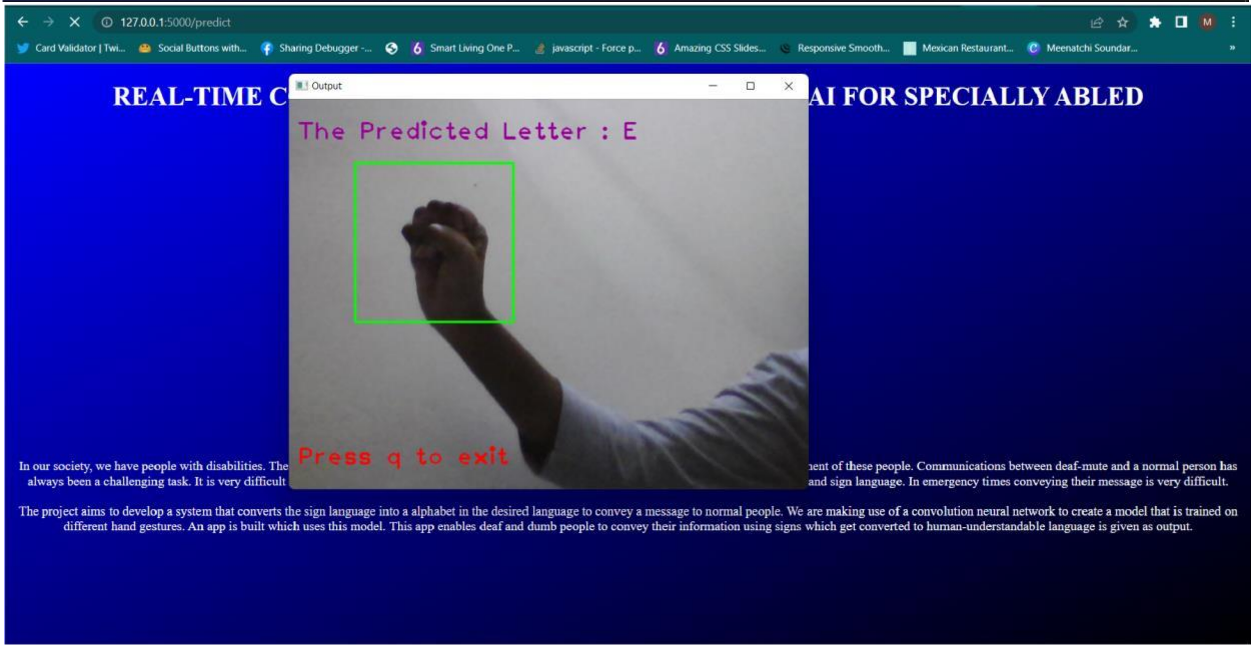
REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

The Predicted Letter : D

Press q to exit

In our society, we have people with disabilities. The technology … e people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute p … nguage. In emergency times conveying their message is very difficult.

The project aims to develop a system that converts the sign language into a alphabet in the desired language to convey a message to normal people. 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 is given as output.



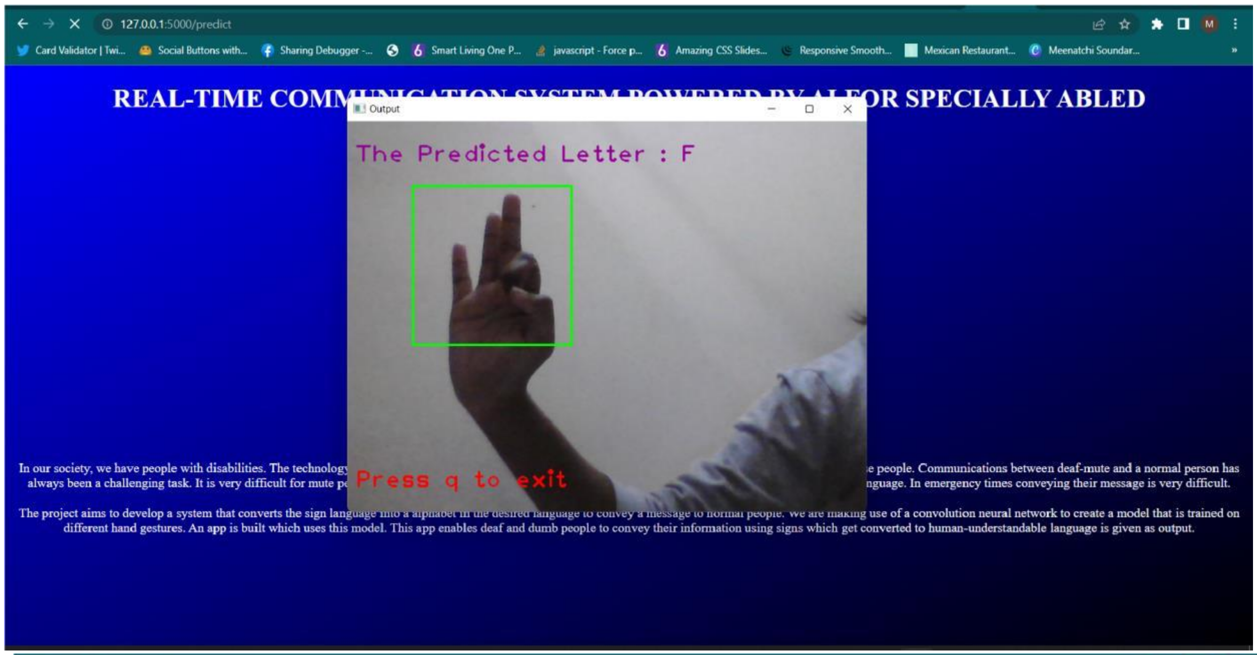
REAL-TIME C … AI FOR SPECIALLY ABLED

The Predicted Letter : E

Press q to exit

In our society, we have people with disabilities. The … nent of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult … and sign language. In emergency times conveying their message is very difficult.

The project aims to develop a system that converts the sign language into a alphabet in the desired language to convey a message to normal people. 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 is given as output.



127.0.0.1:5000/predict

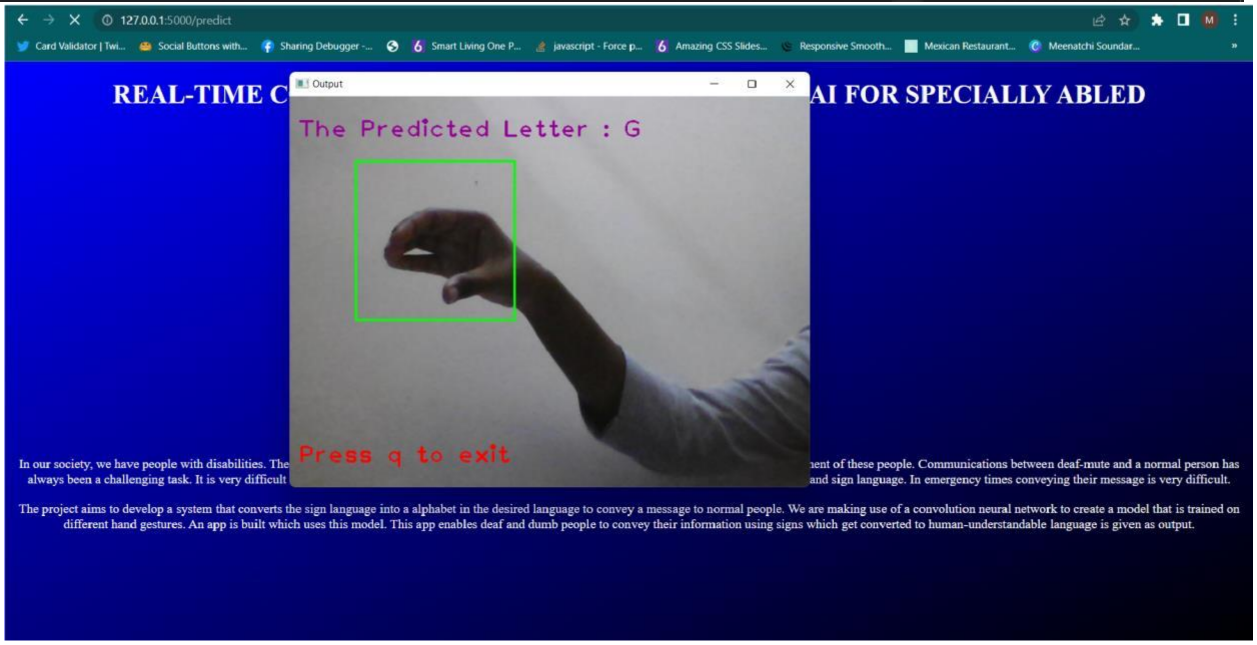
REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

The Predicted Letter : F

Press q to exit

In our society, we have people with disabilities. The technology e people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute pe nguage. In emergency times conveying their message is very difficult.

The project aims to develop a system that converts the sign language into a alphabet in the desired language to convey a message to normal people. 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 is given as output.



127.0.0.1:5000/predict

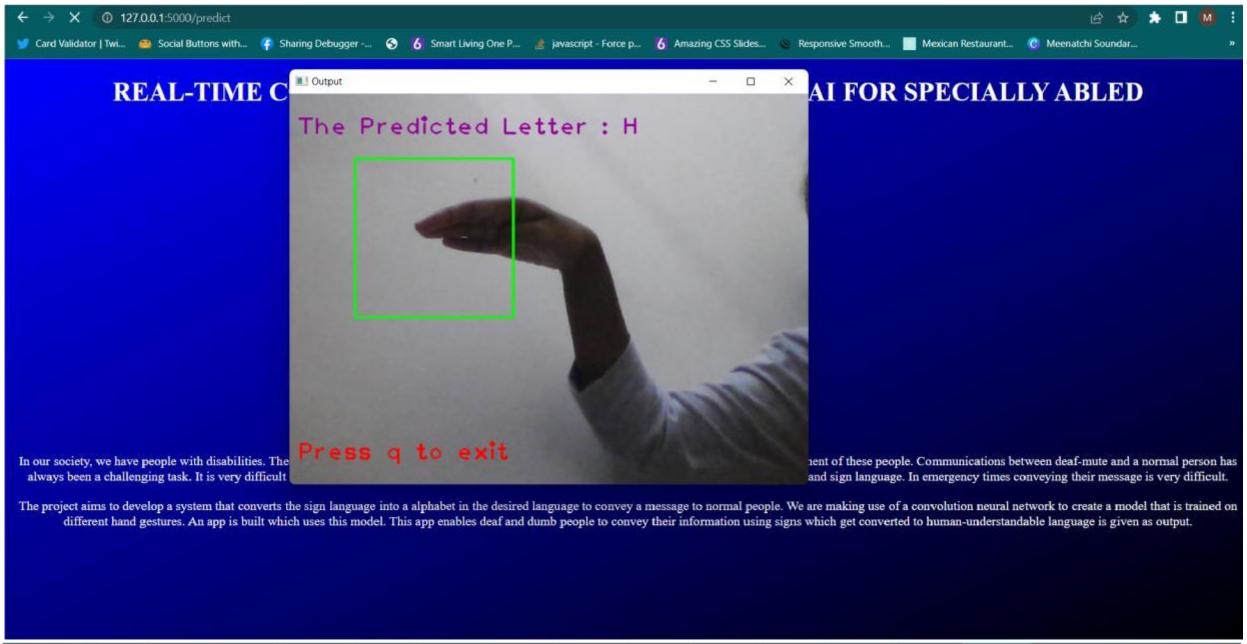
REAL-TIME C AI FOR SPECIALLY ABLED

The Predicted Letter : G

Press q to exit

In our society, we have people with disabilities. The nent of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult and sign language. In emergency times conveying their message is very difficult.

The project aims to develop a system that converts the sign language into a alphabet in the desired language to convey a message to normal people. 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 is given as output.



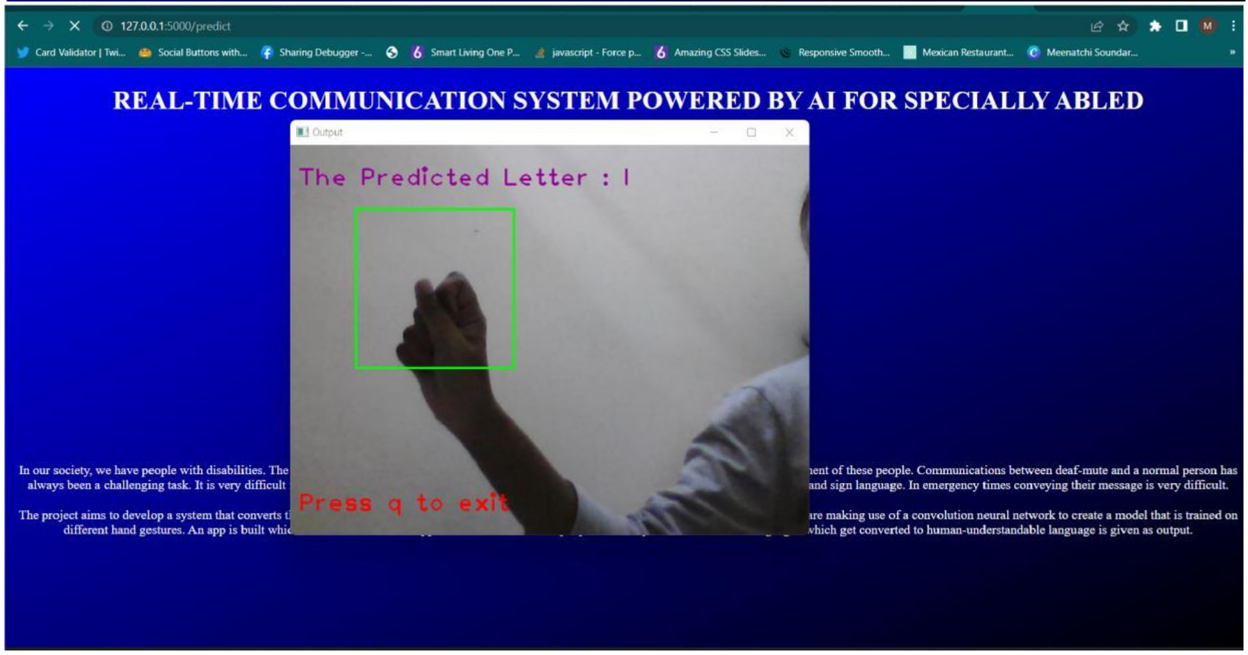
REAL-TIME C… AI FOR SPECIALLY ABLED

The Predicted Letter : H

Press q to exit

In our society, we have people with disabilities. The … nent of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult … and sign language. In emergency times conveying their message is very difficult.

The project aims to develop a system that converts the sign language into a alphabet in the desired language to convey a message to normal people. 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 is given as output.



REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

The Predicted Letter : I

Press q to exit

In our society, we have people with disabilities. The … nent of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult … and sign language. In emergency times conveying their message is very difficult.

The project aims to develop a system that converts t… are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built whic… which get converted to human-understandable language is given as output.

CHAPTER 10

ADVANTAGE AND DISADVANTAGE

ADVANTAGE:

- Communication is the key in this society people with disability tends suffer but the proposed system provides a solution to them.
- Makes the translation of sign language to English easy.
- It can identify and translate the live and moving images.
- The proposed system ensures the easy translation of sign language to English.
- Even the people with lack of sign language can use the proposed system easily.
- This does not require high-end device to use it.
- Can be used on almost all operating systems and browses. ● Does not require prior programming knowledge t use the system
- The proposed system is user friendly.

- Makes the life of the person with disability easy.

DISADVANTAGE:

- The proposed system is not a two-way translation system.
- There is chance for wrong translation.
- Since it is a webpage-based system, it does require internet connectivity which can be inconvenient at times.
- It would have been convenient if it is application based.

CHAPTER 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 recognizes 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.

CHAPTER 12

FUTURE SCOPE

In the future to take the project to the next level two way communication system such as sign language to english and english to sign language is beign under the planning phase.The application version of the web page for both ios and android is also in planning process for the future development.Research to improve the accuracy of the system is under progress.

CHAPTER 13

APPENDIX

SOURCE CODE:

HTML:

```
 <!DOCTYPE html>

 <html>

 <head>

 <title>Real Time Communication</title>

 <style>   body{   background-image: linear-gradient(to bottom

right, blue, black);   background-repeat: no-repeat;

background-attachment: fixed;

 }

 h1,h2,a,p{

color:white;

 }

 </style>
 </head>

 <body>

 <div class="title">

 <h1><center>

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI

FOR SPECIALLY ABLED</center></h1>

</div>

<center><img src="../static/img/img.png" width="300" height="300"></center>  <div>

<center><h2>Show these Gestures to get the Alphabet</h2></center>

</div>

<div>

<center><a href="{{ url_for('predict') }}">CLICK HERE TO SHOW YOUR
GESTURES</a></center>

</div>

<div>

<center>  <p>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.<br>

<br>

The project aims to develop a system that converts the sign language into a alphabet in
the desired language to convey a message to normal people. 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 is
given as output.</p>

</center>  </div>

</body>

</html> PYTHON:  from flask import

Flask,render_template,request import cv2  from

keras.models import load_model import numpy

as np from gtts import gTTS import os  from

keras.preprocessing import image from

skimage.transform import resize from playsound

import playsound app = Flask(_name_)

model=load_model("aslpng1.h5")

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

@app.route('/', methods=['GET'])

def index():

          return

render_template('index.html')

@app.route('/index', methods=['GET']) def home():

          return     render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])

def predict():
```

```
                        print("[INFO] starting video
stream...")                         vs = cv2.VideoCapture(0)
(W, H) = (None, None)                          while True:
            (grabbed, frame) = vs.read()
                                  if not grabbed:
break                                       if W is None or H
is None:
                                                (H, W) = frame.shape[:2]
output = frame.copy()
                                    # r = cv2.selectROI("Slect", output)
                                # print(r)
cv2.rectangle(output, (81, 79), (276,274), (0,255,0), 2)
frame = frame[81:276, 79:274]                                     frame =
cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
                                    _, frame = cv2.threshold(frame, 95, 255,
cv2.THRESH_BINARY_INV)
                                    frame = cv2.cvtColor(frame,
cv2.COLOR_GRAY2RGB)                                     img =
resize(frame,(64,64,3))                                    img =
np.expand_dims(img,axis=0)                                      if(np.max(img)>1):
                                                img          =          img/255.0
result                =                  np.argmax(model.predict(img))
```

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

result=str(index[result])

                             cv2.putText(output, "The Predicted Letter : {}".format(result), (10,
50), cv2.FONT_HERSHEY_PLAIN,

                                                                      2, (150,0,150), 2)

                                      cv2.putText(output, "Press q to exit", (10,450),
cv2.FONT_HERSHEY_PLAIN, 2, (0,0,255), 2)

                             speech = gTTS(text = result, lang = 'en', slow = False)

cv2.imshow("Output", output)                                    key =

cv2.waitKey(1) & 0xFF                                   if key == ord("q"):

                                          break

print("[INFO] cleaning up...")

vs.release()                          cv2.destroyAllWindows()

return render_template("index.html") if

_name_ == '_main_':

              app.run(debug=True)

TRAINNING CODE:  # Importing Libraries  from

tensorflow.keras.preprocessing.image import ImageDataGenerator  #

Image Augmentation train_datagen = ImageDataGenerator(rescale =

1./255, shear_range = 0.2, zoom_range =

0.2, horizontal_flip = True) test_datagen =

ImageDataGenerator(rescale = 1./255)

# Loading train and test set
```

```
X_train = train_datagen.flow_from_directory(r"D:\Real-time Communication System for
specially abled\Dataset\training_set", target_size = (64, 64), batch_size = 32, class_mode
= 'categorical')

X_test = test_datagen.flow_from_directory(r"D:\Real-time Communication System for
specially abled\Dataset\training_set", target_size = (64, 64), batch_size = 32, class_mode
= 'categorical')

# checking indices

X_train.class_indices # Importing Libraries  from tensorflow.keras.models

import Sequential from tensorflow.keras.layers import Dense  from

tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten

# Initializing the Model model = Sequential() # Adding Convolution Layer

model.add(Convolution2D((32), (3,3), input_shape = (64, 64, 3), activation =

'relu'))

# Adding Pooling Layer

model.add(MaxPooling2D(pool_size = (2, 2)))  # Adding

Flatten Layer model.add(Flatten()) # Adding Hidden Layer

model.add(Dense(units = 512, kernel_initializer =

'random_uniform', activation = 'relu'))

# Adding Output Layer model.add(Dense(units = 9, kernel_initializer =

'random_uniform', activation = 'softmax')) # Compile the model

model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics =

['accuracy']) # Fiiting the model model.fit_generator(X_train, steps_per_epoch =

24, epochs = 10, validation_data = X_test, validation_steps = 40) # Saving the model

model.save('aslpng1.h5')
```

TESTING CODE:

```
# Importing Libarries  from
tensorflow.keras.models import load_model from
tensorflow.keras.preprocessing import image
import numpy as np import cv2 # loading model
model = load_model('aslpng1.h5') from
skimage.transform import resize def detect(frame):
   img = resize(frame, (64, 64, 3))
img = np.expand_dims(img, axis = 0)
if np.max(img) > 1:         img =
img/255.0      prediction =
model.predict(img)
print(prediction)

   return prediction frame = cv2.imread(r"D:\Real-time Communication
System for specially abled\Dataset\test_set\A\16.png") data = detect(frame)
index = ['A','B','C','D','E','F','G','H','I']
index[np.argmax(data)] # Importing Libraries
import cv2 import numpy as np  from
tensorflow.keras.models import load_model from
tensorflow.keras.preprocessing import image
# Loading Model model =
load_model("aslpng1.h5") video =
cv2.VideoCapture(0)
```

```
index = ['A','B','C','D','E','F','G','H','I'] while

True:

   success, frame = video.read()

cv2.imwrite('frame.jpg', frame)     img =

image.load_img('frame.jpg', target_size = (64, 64))     x =

image.img_to_array(img)     x = cv2.cvtColor(x,

cv2.COLOR_BGR2HSV)     a = x.array_to_img(x)

cv2.imshow("")  x = np.expand_dims(x, axis = 0)     pred =

np.argmax(model.predict(x), axis = 1)

y = pred[0]     copy = frame.copy()     cv2.rectangle(copy,

(320, 100), (620, 400), (255, 0, 0), 5)

   cv2.putText(frame, "The Predicted Alphabet : " + str(index[y]), (100, 100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)

cv2.imshow('frame',    frame)                 if

cv2.waitKey(1) & 0xFF == ord('q'):

      break


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

LINK:

https://github.com/IBM-EPBL/IBM-Project-50789-1660924149**DEMO** l

LINK:

https://drive.google.com/file/d/16×178uOpx3uahe1p9osFIh10NE_y3s3G/view?
Usp=share_link