

**REAL-TIME COMMUNICATION SYSTEM POWERED  
BY AI FOR SPECIALLY ABLED  
A PROJECT REPORT**

Submitted by

**TEAM ID     PNT2022TMID25959**

**TEAM LEADER:   ARUMUGAM RAJKUMAR**

**TEAM MEMBER 1: GNANESHWAR REDDY**

**TEAM MEMBER2: BEDATHALA HEMANTHKUMAR**

**TEAM MEMBER 3: P UDAY KIRAN**

**BACHELOR OF ENGINEERING**

in

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**PANIMALAR INSTITUTE OF TECHNOLOGY**

**ANNA UNIVERSITY CHENNAI : 600025**

# CONTENTS

## **1. INTRODUCTION**

- 1.1 Project Overview
- 1.2 Purpose

## **2. LITERATURE SURVEY**

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

## **3. IDEATION & PROPOSED SOLUTION**

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

## **4. REQUIREMENT ANALYSIS**

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

## **5. PROJECT DESIGN**

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

## **6. PROJECT PLANNING & SCHEDULING**

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

## **7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

- 7.1 Feature 1
- 7.2 Feature 2

## **8. TESTING**

- 8.1 Test Cases
- 8.2 User Acceptance Testing

## **9. RESULTS**

- 9.1 Performance Metrics

## **10. ADVANTAGES & DISADVANTAGES**

## **11. CONCLUSION**

## **12. FUTURE SCOPE**

## **13. APPENDIX**

Source Code  
GitHub & Project Demo Link

<b>TITLE</b>	REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED
<b>TEAM ID</b>	<b>PNT2022TMID25959</b>
<b>TEAM LEAD</b>	ARUMUGAM RAJKUMAR
<b>TEAM MEMBERS</b>	GNANESHWAR REDDY  BDATHALA HEMANTHKUMAR  P UDAY KIRAN

## **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:

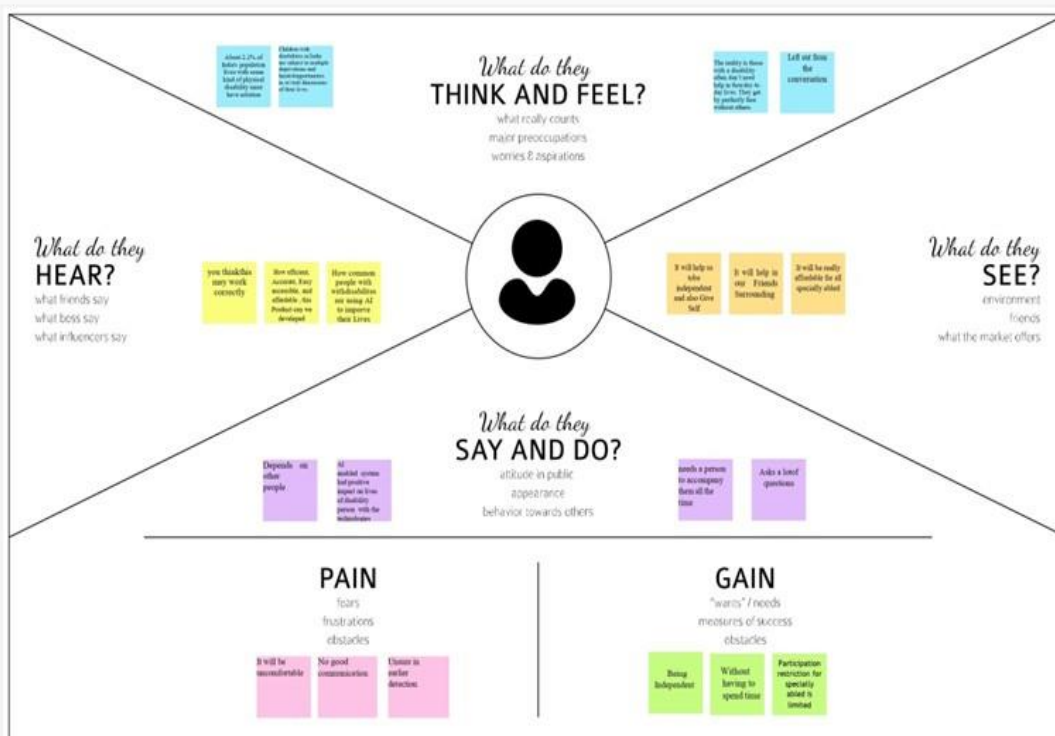


Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Person without disability	Communicate with deaf and dumb via phone	It is not possible	I can't able to understand the sign language	Frustrated
PS-2	A human like everyone	Communicate freely with others	I cannot do so	I am a deaf/dumb	Captivated as well as unmotivated

## CHAPTER 3

### IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas





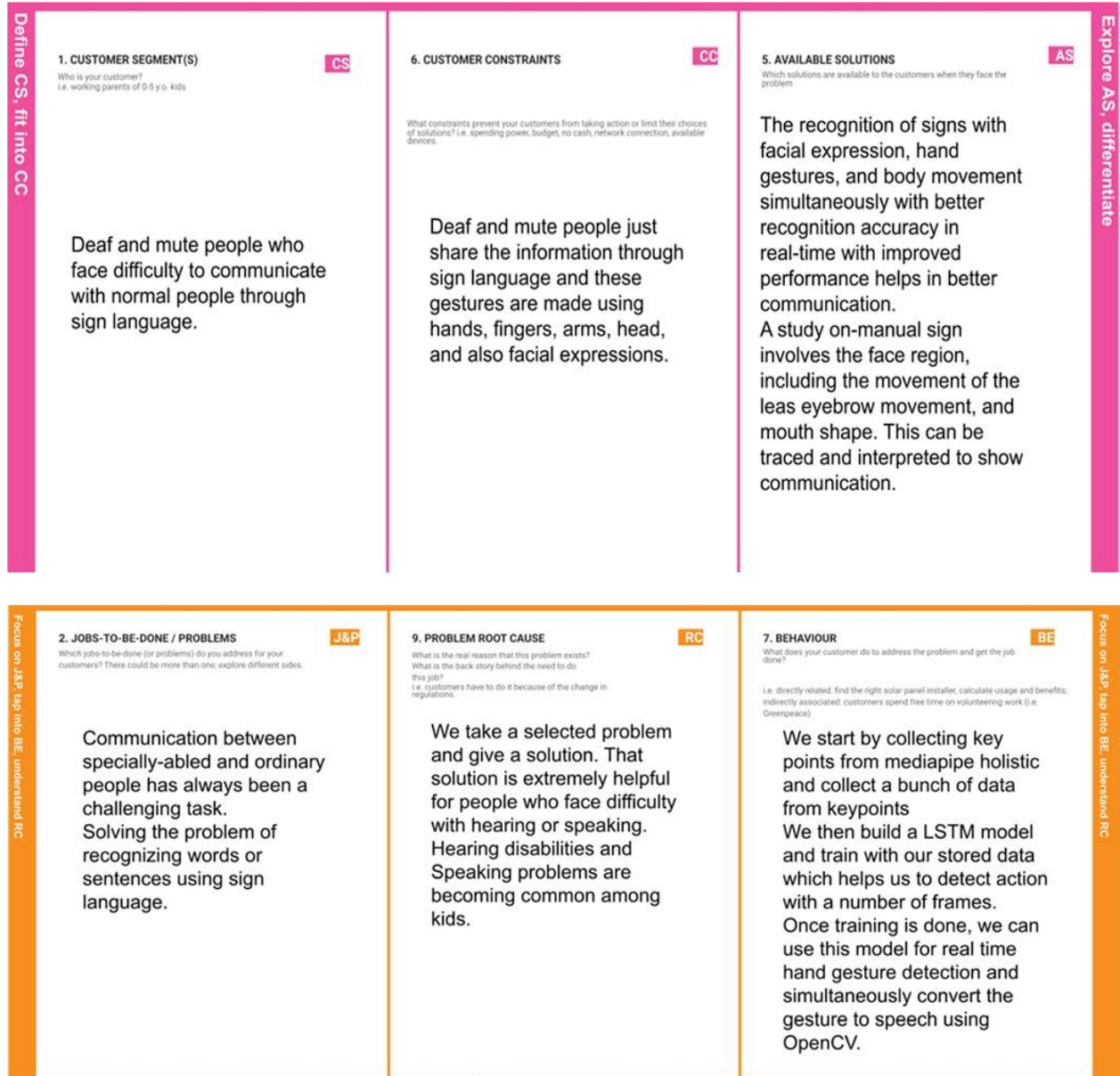
### 3.2 Ideation & Brainstorming

[illegible]

### 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 known by everyone, people with disabilities tend 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-based sign 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 with every gesture possible.
4.	Social Impact/ Customer Satisfaction	<ol style="list-style-type: none"> <li>1. To boost the confidence of a differently abled person by making them independent</li> <li>2. To break the communication barrier</li> <li>3. Ease the conversion of sign language to voice-based message</li> <li>4. To improve the everyday lives of people with disabilities</li> </ol>
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 around the globe.

### 3.4 Problem Solution fit



Identify strong TR & EM	<b>3. TRIGGERS</b> <b>TR</b> 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.	<b>10. YOUR SOLUTION</b> <small>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.</small> <b>SL</b>  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.	<b>8. CHANNELS of BEHAVIOUR</b> <b>CH</b> <b>8.1 ONLINE</b> What kind of actions do customers take online? Extract online channels from #7.  <b>8.2 OFFLINE</b> 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
	<b>4. EMOTIONS: BEFORE / AFTER</b> <b>EM</b> <small>How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure &gt; confident, in control - use it in your communication strategy &amp; design.</small>  The emotions are frustrated, anger, left out, lonely, fear, neglected			

## 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

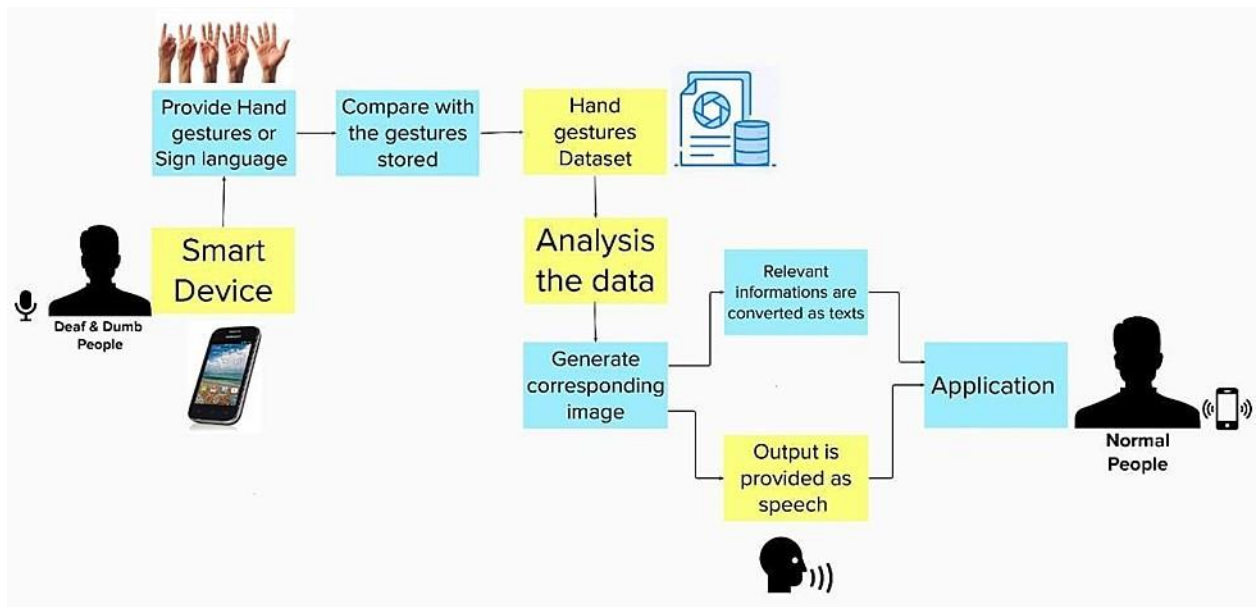
##### Software Requirements

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

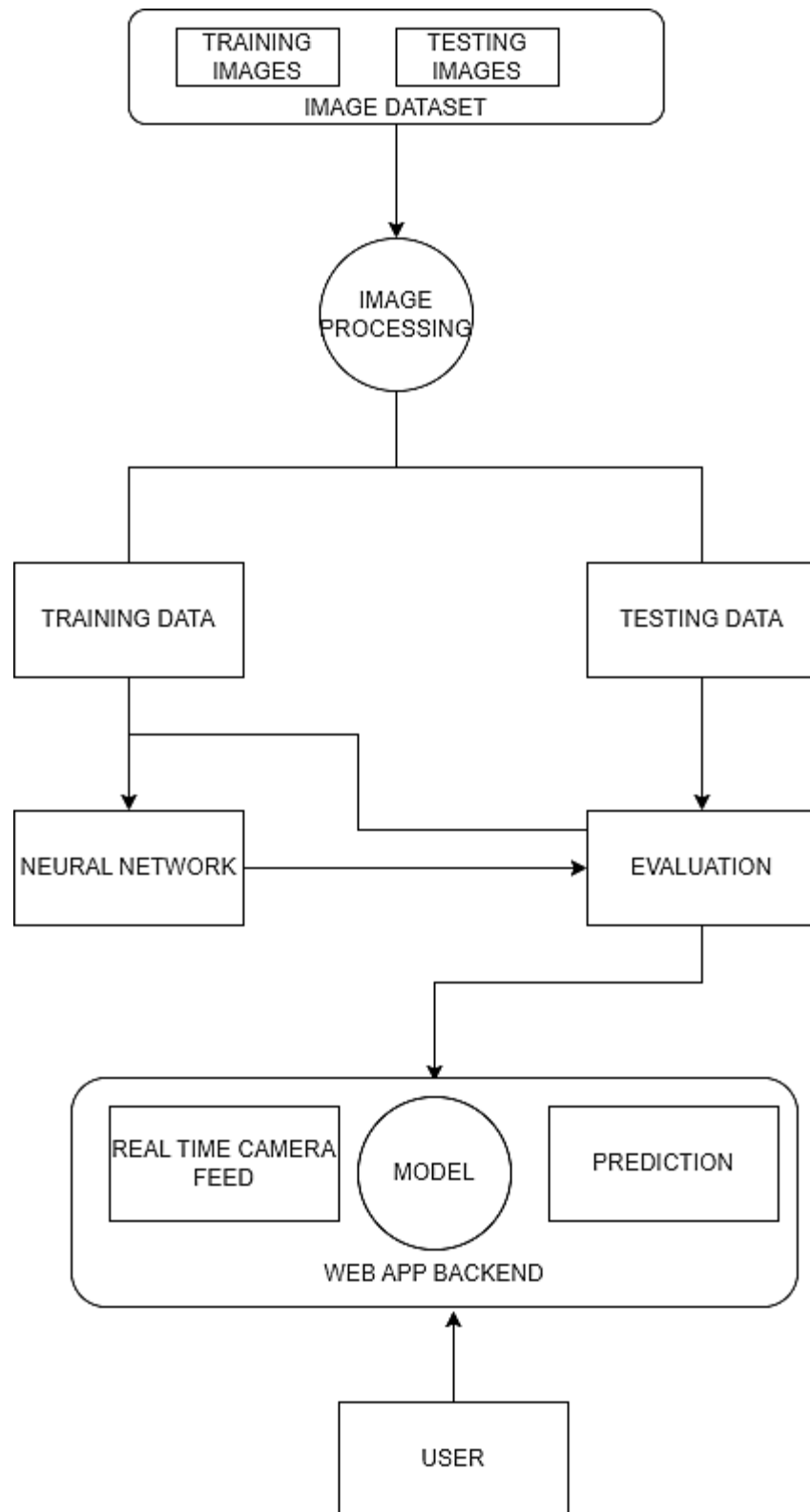
## CHAPTER 5

### PROJECT DESIGN

#### 5.1 Data Flow Diagrams



**Data Flow Diagram**



**Flow Chart**

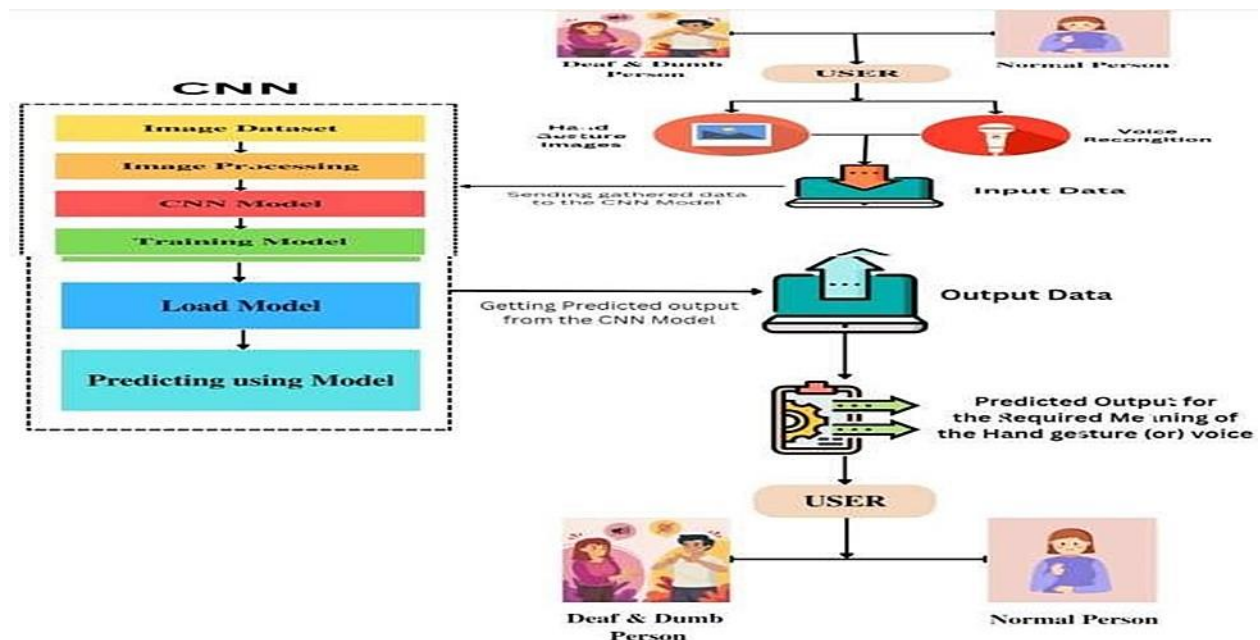
## 5.2 Solution & Technical Architecture

Solution Architecture:

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

- The best tech solution to solve existing business problems.
- Describing the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Defined about features, 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	Functional Requirement( Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Register with the users information.	USN-1	As a user, I can register for the application by entering my email,password, and confirming my password.	I can access my account / dashboard in the application.	High	Sprint-1
Customer (Deaf people)	To communicate with people using signs.	USN-2	As a user, I can see my application and made changes in any browser and register to it.	I can login and see my account in the application anywhere at any time.	High	Sprint-1
Customer (Dumb people)	To communicate with people easily and efficiently.	USN-3	As a user, I can see my application and made changes in any browsers and register to it.	I can login and see my account in the application 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 application by entering my email, password, and confirmation is made.	I can login and see my account.	Medium	Sprint-2
Customer (Learner of Sign language)	User needs to be aware and learn about sign language.	USN-5	As a user, I can create my account in the application with my email and password, to get knowledge about sign languages.	I can create my account and access the dashboard in the application.	High	Sprint-1
Customer (Web user)	They want the update on the application condition.	USN-6	As a user, I can register for the application by entering my email, password, and confirming my password. To get details about real-time communication.	I can able to use any browser to access the application from anywhere, to know anything about real-time communication.	High	Sprint-1
Customer Care Executive	They want to help people by sending application conditions.	USN-7	As a user, I can receive a message from the administration about conditions of application of real-time communication.	I will analyse and send SMS to the people.	High	Sprint-1

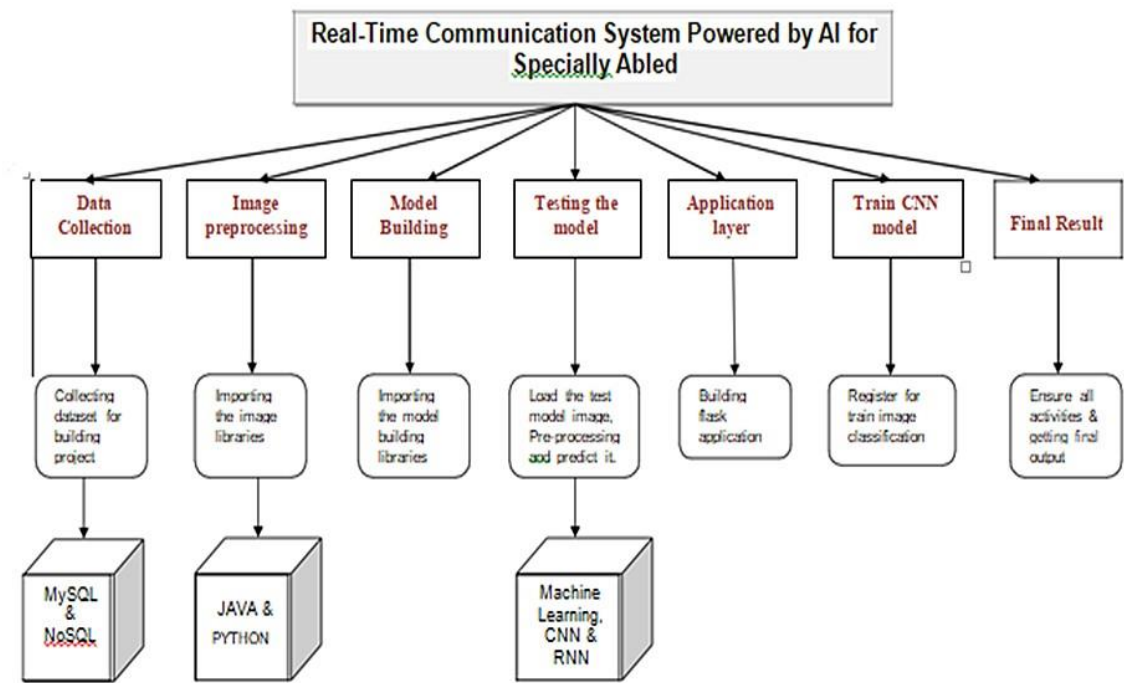
## CHAPTER 6 PROJECT PLANNING AND SCHEDULING

### 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	Train Conversational Engine	M6	Register for IBM Cloud and train Image Classification Model
Milestone 7	Final Result	M7	To ensure all the activities and resulting the final output.

**MILESTONE ACTIVITY PLAN**



## 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 the dataset	8	Medium
Sprint – 2	Model Building	USN – 3	Import the required libraries, add the necessary layers and compile the model	10	High
Sprint – 2		USN – 4	Training the image classification model using CNN	7	Medium
Sprint – 3	Training and Testing the Model	USN – 5	Training the model and testing the model's performance	9	High
Sprint – 4	Applica on Development	USN – 6	Conver ng the input gesture image into English Alphabets	8	Medium

## 6.2 Sprint Delivery Schedule

<b>Sprint</b>	<b>Total StoryPoints</b>	<b>Duration</b>	<b>Sprint StartDate</b>	<b>Sprint End Date (Planned)</b>	<b>Story Points Completed (as on Planned End Date)</b>	<b>Sprint Release Date (Actual)</b>
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:  $AV = 17 \div 6 = 2.83$**

**2. Sprint – 2:  $AV = 17 \div 6 = 2.83$  '**

**3. Sprint – 3:  $AV = 9 \div 6 = 1.5$**

**4. Sprint – 4:  $AV = 5 \div 6 = 0.83$**

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></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 Libraries 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

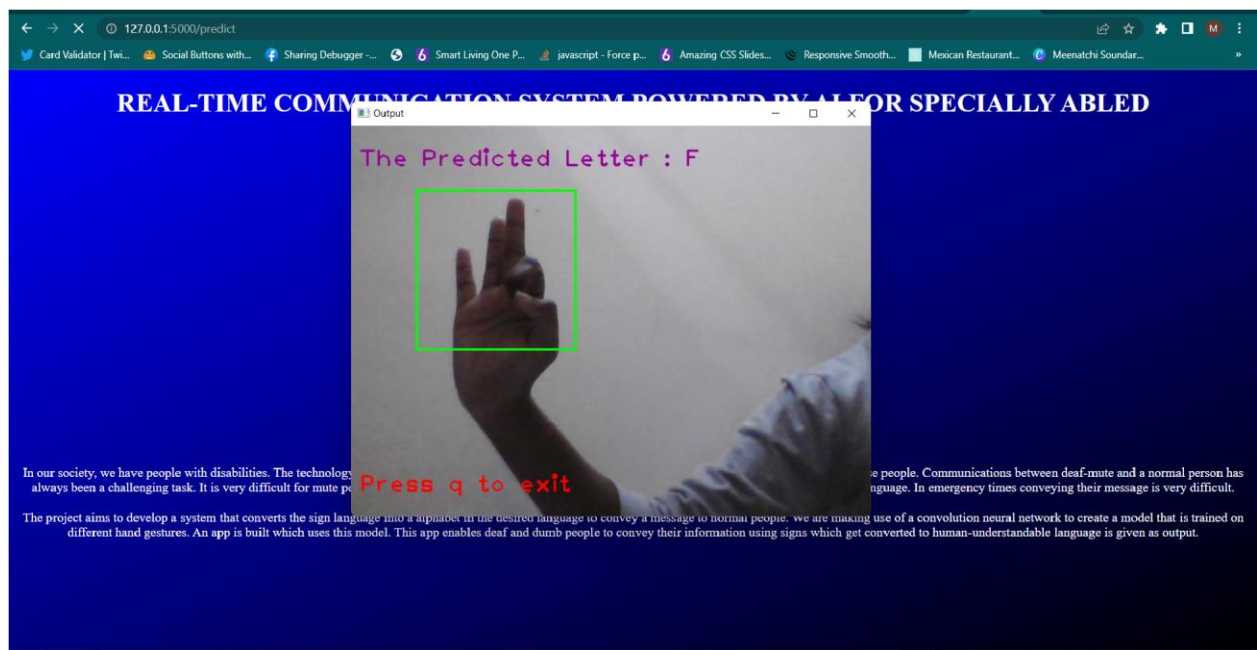
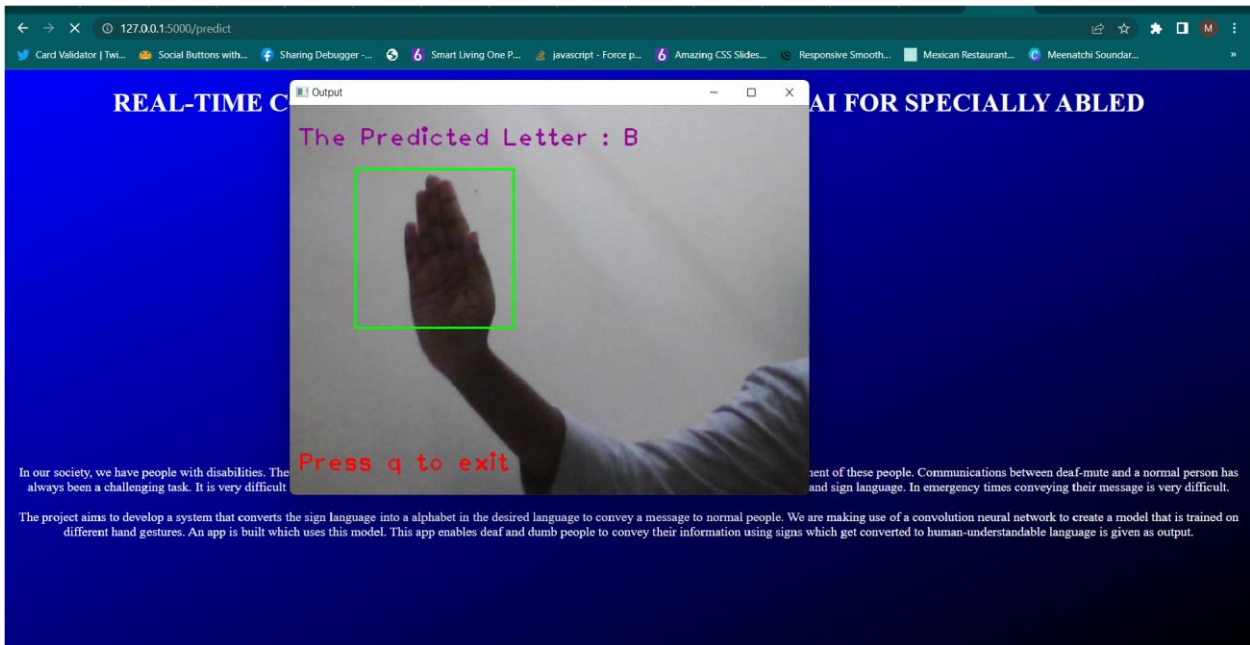
```

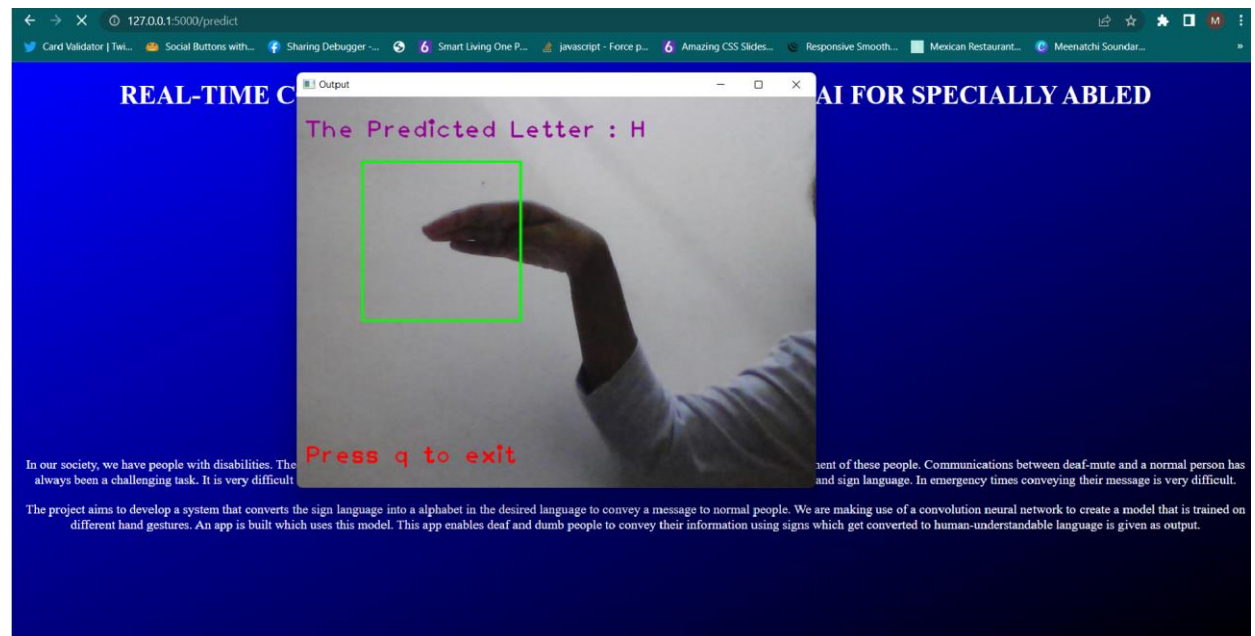
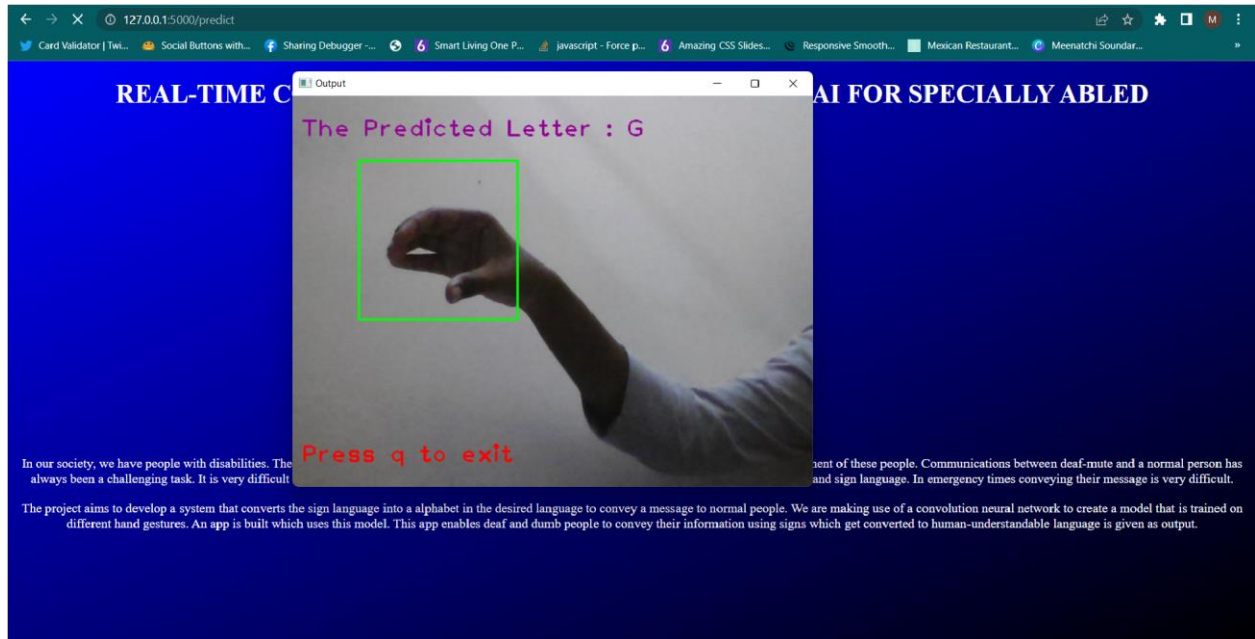
```
video.release() cv2.destroyAllWindows()
```

## **CHAPTER 9**

### **RESULT**

#### **9.1 Performance Metrics**







## **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></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 Libraries

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/IBMEPBL/IBMProje>

[ct-2090-1658428100](#)

**LINK :[DEMO](#)**