Final Report

A Real-Time Communication System For Specially Abled

TERM ID : PNT2022TMID28501

Team Members:

1. Gowtham Kannan R
2. Chithrai Selvan
3. Kannikhan Choudry
4. Arun Kumar M

TABLE OF CONTENTS

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

- 7.1 Feature 1
- 7.2 Feature 2
- 7.3 Database Schema (if Applicable)

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

1. Introduction 2. Literature Survey

1.1 Project Overview,

The project deals with building an application that helps typically challenged people to communicate between themselves and the common people. Communication between a person with hearing/speech impairment and a normal person has always been a challenging task. This application tries to reduce the barrier of communication by developing an assistive application for specially challenged people To have 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.

1.2 Purpose

The project intends to create a system that can translate speech into specified sign language for the deaf and dumb as well as translate sign language into a humanhearing voice in the desired language to communicate a message to normal people. A convolution neural network is being used to build a model that is trained on various hand motions. Based on this model, an app is created. With the help of this app, persons who are deaf or dumb can communicate using signs that are translated into speech and human-understandable words.

2. LITERATURE SURVEY

2.1 Existing Problem

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 complicated 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 handy to have a proper conversation between a normal person and an impaired person in any language.

2.2 References

Text to speech conversion • S. Venkateswarlu The present paper has introduced an innovative, efficient and realtime cost beneficial technique that enables user to hear the contents of text images instead of reading through them. It combines the concept of Optical Character Recognition (OCR) and Text to Speech Synthesizer (TTS) in Raspberry pi. This device consists of two modules, image processing module and voice processing module. The device was developed based on Raspberry Pi v2 with 900 MHz processor speed. • Easy get hacked • Less accurate Design of the architecture for text recognition and reading in an online assessment applied to visually impaired students • Alex Leon This paper describes the architecture for text recognition and reading in an online assessment applied to visually impaired students. For this purpose, it is intended to implementation online evaluation system exclusively to recognize alphanumeric information, i.e., letters and numbers, through the use of an Application Programming Interface or also known as speech and text processing API's, where the computer can understand and respond in natural language • Operating system Problem. • Chance of misunderstanding Voice source modelling using deep neural networks for statistical parametric speech synthesis • Tuomo Raitio A voice source modelling method employing a deep neural network (DNN) to map from acoustic features to the time-domain glottal flow waveform. First, acous-tic features and the glottal flow signal are estimated from each frame of the speech database. Pitch-synchronous glottal flow time-domain

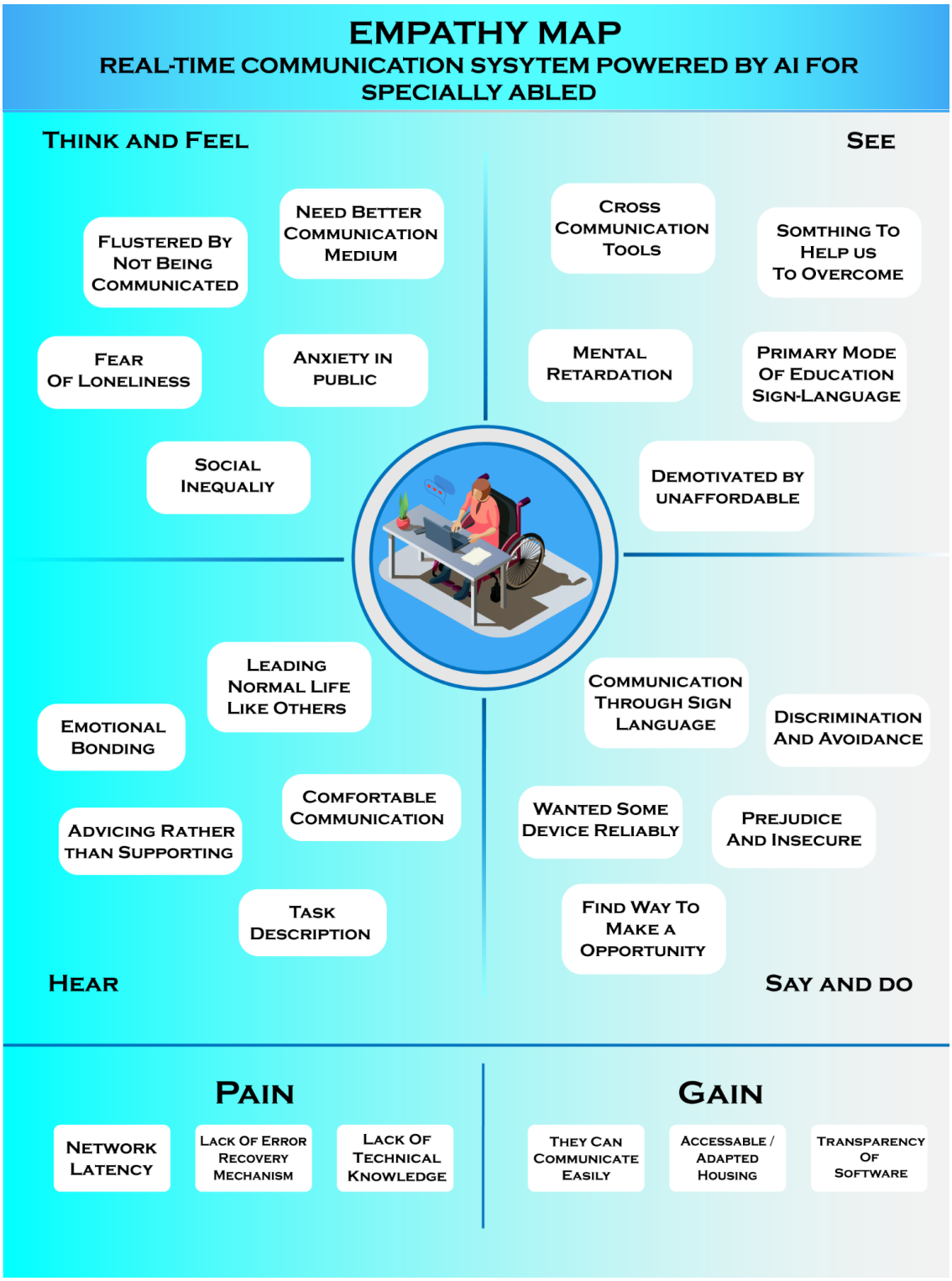
waveforms are extracted, interpolated to a constant duration, and stored in a codebook. Then, a DNN is trained to map from acoustic features to these duration-normalised glottal waveforms. At synthesis time, acoustic features are generated from a statistical parametric model, and from these, the trained DNN predicts the glottal flow wave-form. • High implementation costs. • Noisy environment 2.3 Problem Definition Statement People with disabilities are a part of our society. Even though technology is constantly evolving, little is being done to improve the lives of these people. Communication with a deaf-mute person has always been difficult. Because hand sign language is not taught to the general public, it can be difficult for silent people to communicate with non-mute people. In times of crisis, they may find it difficult to communicate. When other modes of communication, such as speech, are unavailable, the human hand has remained a popular method of information transmission. A voice conversion system with hand gesture recognition and translation will be very helpful in establishing proper communication between a normal person and a handicapped person in any language.

2.3 Problem Definition Statement

People with disabilities are a part of our society. Even though technology is constantly evolving, little is being done to improve the lives of these people. Communication with a deaf-mute person has always been difficult. Because hand sign language is not taught to the general public, it can be difficult for silent people to communicate with non-mute people. In times of crisis, they may find it difficult to communicate. When other modes of communication, such as speech, are unavailable, the human hand has remained a popular method of information transmission. A voice conversion system with hand gesture recognition and translation will be very helpful in establishing proper communication between a normal person and a handicapped person in any language.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

Template

Brainstorm & Idea Priortization

Executing a brainstorm isn't unique; holding a productive brainstorm is. Great brainstorms are ones that set the stage for fresh and generative thinking through simple guidelines and an open and collaborative environment. Use this when you're just kicking-off a new project and want to hit the ground running with big ideas that will move your team forward.

15 minutes to prepare
30-60 minutes to collaborate
3-8 people recommended

Share template feedback

Before you collaborate

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

15 minutes

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

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

C **Learn How To Use The Facilitation Tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open 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.

8 minutes

Problem Statement

Real-Time Communication System Powered By AI For Specially Abled

Key rules of brainstorming

To run an smooth and productive session

1. Stay in topic.
2. Encourage wild ideas
3. Defer judgment.
4. Listen to others
5. Go for volume.
6. If possible, be visual

2

Brainstorm solo

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

5 minutes

CHITHRAI SELVAN S

List out the problems the differentially abled person faces day to day	The sign to text/sound translation should be accurate	Implement the system as a web app
The limitations should recognised and addressed	Extract palm orientation as attribute	Use regression as supervised learning technique for training
Transulate email document	The web interface should be convinient for every people	Take normal person input as audio

ARUN KUMAR S

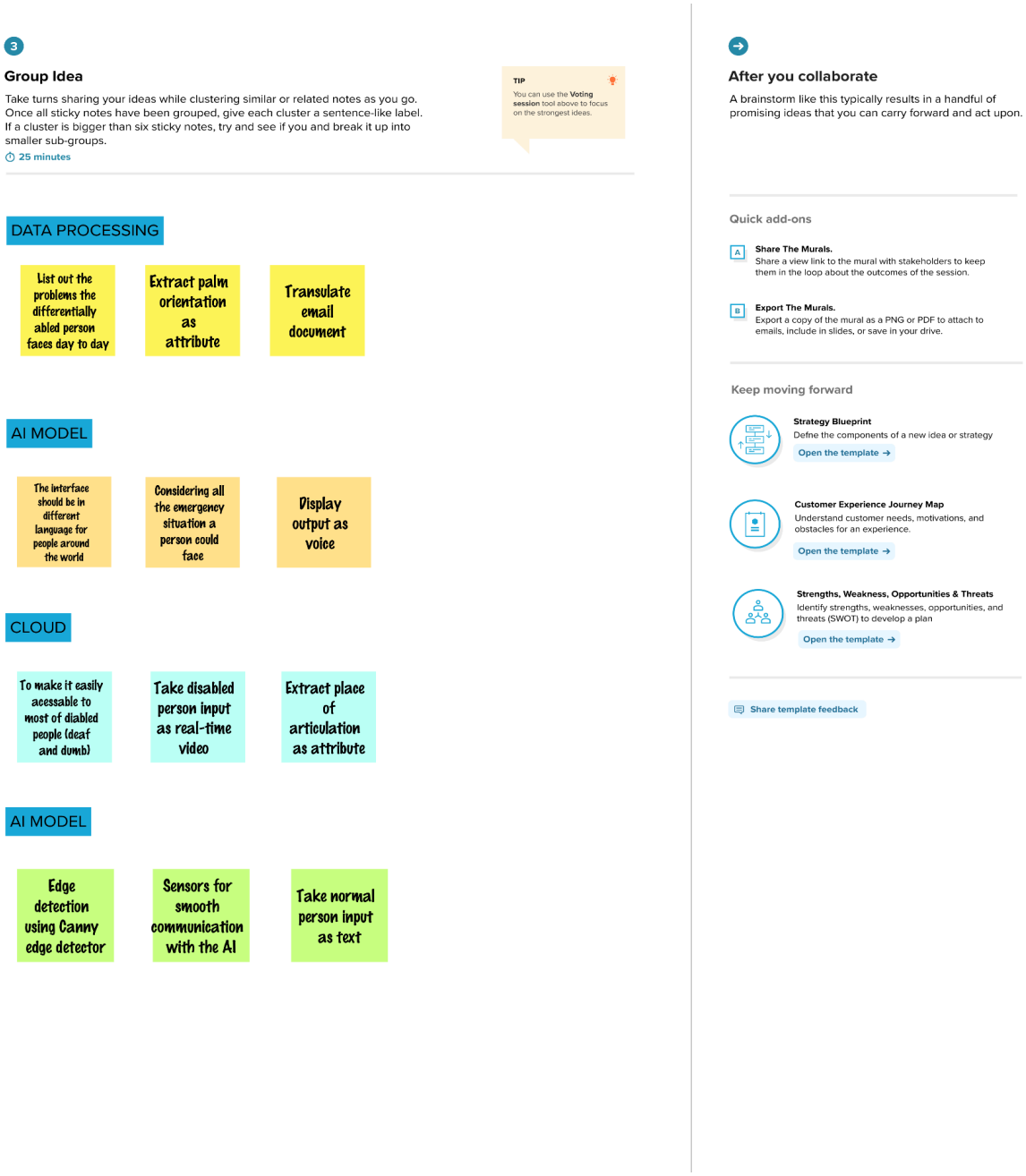
Extract hand shape as parameters	Considering all the emergency situation a person could face	Display output as voice
The interface should be in different language for people around the world	Target sign language - ASL	The data privacy and security must be maintained
Edge detection using Laplacian detector	Considering all the emergency situation a person could face	Extract hand shape as parameters

GOWTHAM KANNAN R

Alert the authorities in case of emergency	Implement the system as a mobile app	Create an moral and emotional support to boost
Edge detection using Canny edge detector	Sensors for smooth communication with the AI	Take normal person input as text
Display output as text	Target sign language - Australian Sign Lang	Easy to be communicate with other people with same or different disability

KANNIKHAN CHOUDRY D

To make it easily acessable to most of diabled people (deaf and dumb)	Make it portable so it can be used anywhere at anytime	It must be cross platform compatible
It should be in a way that it can convey emotions of other people as well as themselves while communicating	Take disabled person input as real-time video	Compare the image with dataset by superimposition
Use classifcation as supervised learning technique for training	It must be cross platform compatible	Extract place of articulation as attribute

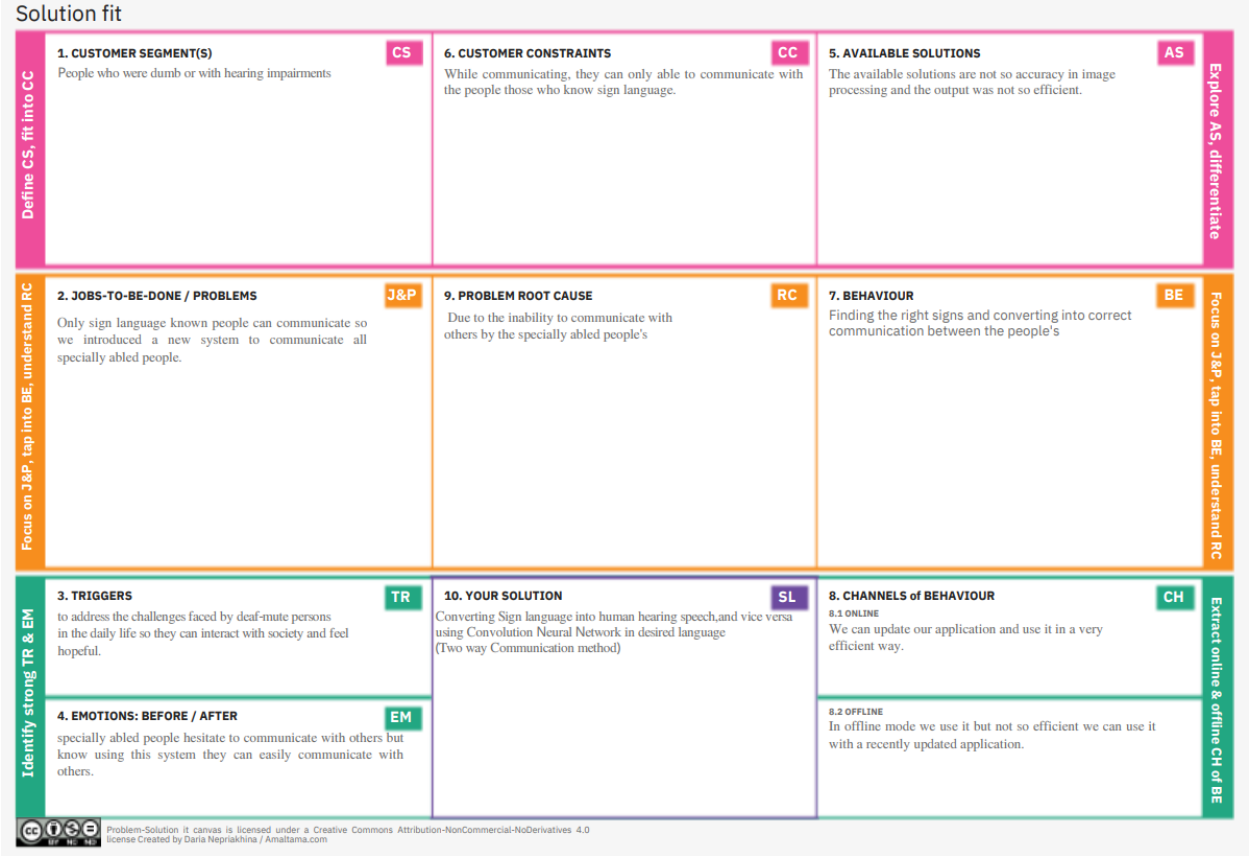


3.3 Proposed Solution

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

S.No.	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	Differently able like dump and mute people can communicate only through the sign language,normal people those who do not know the sign language feels difficult to communicate with them.
2.	Idea / Solution description	To overcome this problem we have an idea that an application is created to communicate with the normal people.
3.	Novelty / Uniqueness	This process the image of the person who is using sign language and convert it into the voice by analyzing the sign used.
4.	Social Impact / Customer Satisfaction	Differently able people feel free to communicate and it bring a huge difference comparing to past.
5.	Business Model (Revenue Model)	There are many people in the world who is differently able,this application will become more popular among them and it will be installed by all and it will be used,and so it will produce more money.
6.	Scalability of the Solution	Thus this would bring a new evolution in Real Time Communication System Powered by AI for Specially Able with less time and safe enough resources.

3.4 Proposed Solution Fit



4.REQUIREMENT ANALYSIS

4.1 Functional Requirement

Following are the functional requirements of the proposed solution.

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

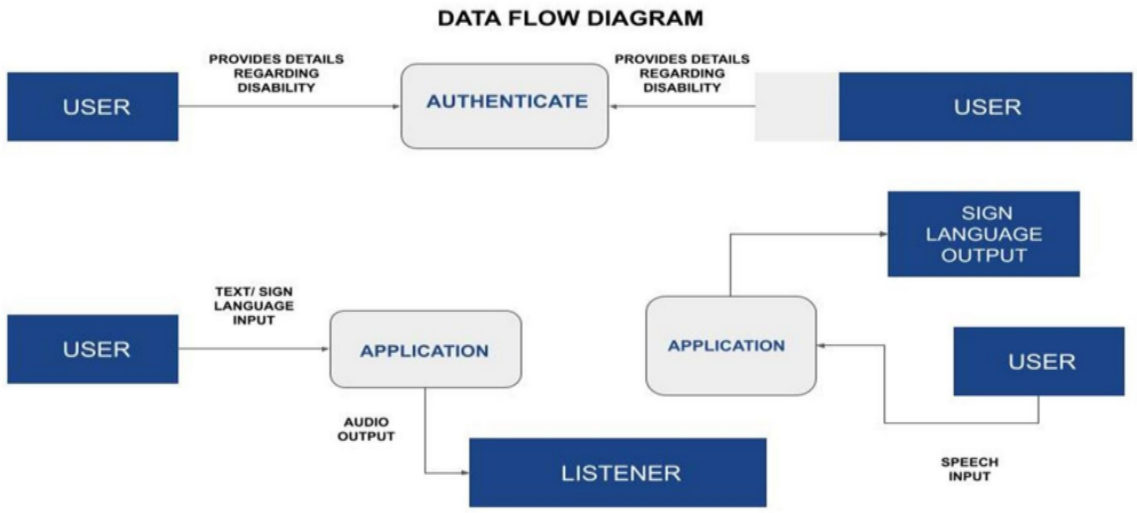
4.2 Non-Functional Requirement

FR No.	Non- Functional Requirement	Description
NFR-1	Usability	The camera captures all expressions including facial expressions and hand gestures which can be easily used by all age groups. It can be used by deaf-mute people and their care takers.
NFR-2	Security & Privacy	The system is more secure and information of the customers is also maintained confidentially.
NFR-3	Accuracy	The system must have a great accuracy rate. The accuracy is important so that the disabled students could get a clear understanding.
NFR-4	Performance	The performance of the model is efficient. The cost-effective nature of the system makes it extremely liable. The latency is very less for the conversion process.
NFR-5	Availability	The solution is suitable for different languages and can be used in many countries. It can be trained for all the available sign languages. This model can be used at any time anywhere.
NFR-6	Scalability	The system gives output rapidly. It also predicts quickly when it gets so many inputs at a time. It predicts different types of sign language at a time. Upto 25000 users can be use this model at a time.

5. PROJECT DESIGN

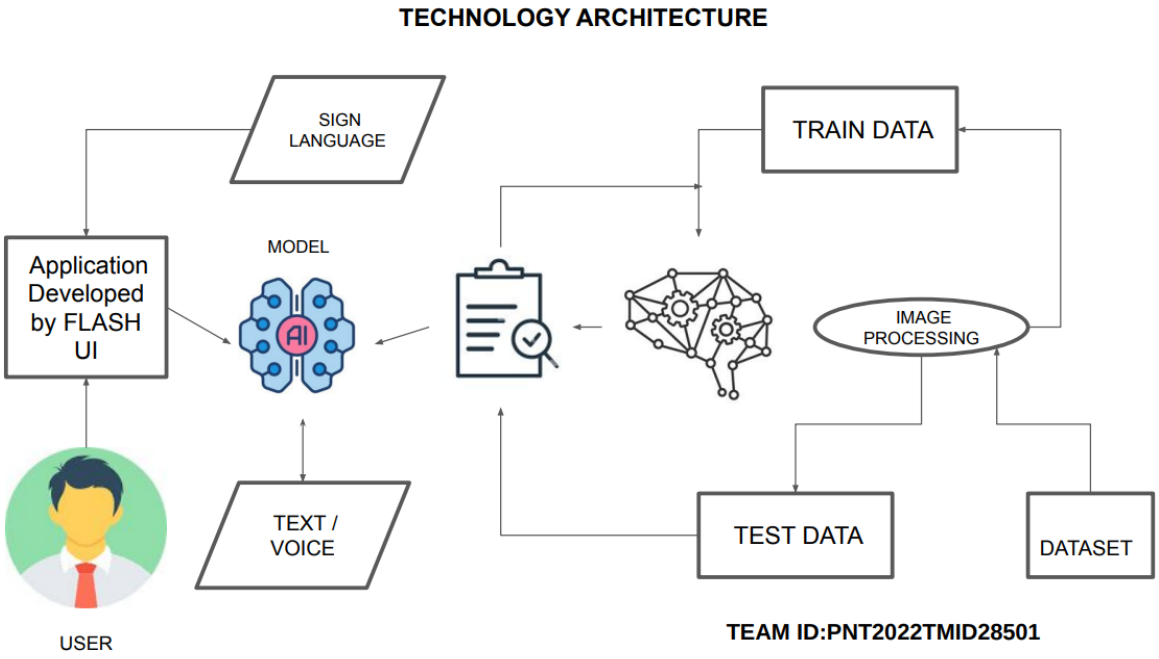
5.1.Flow Data Diagram

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

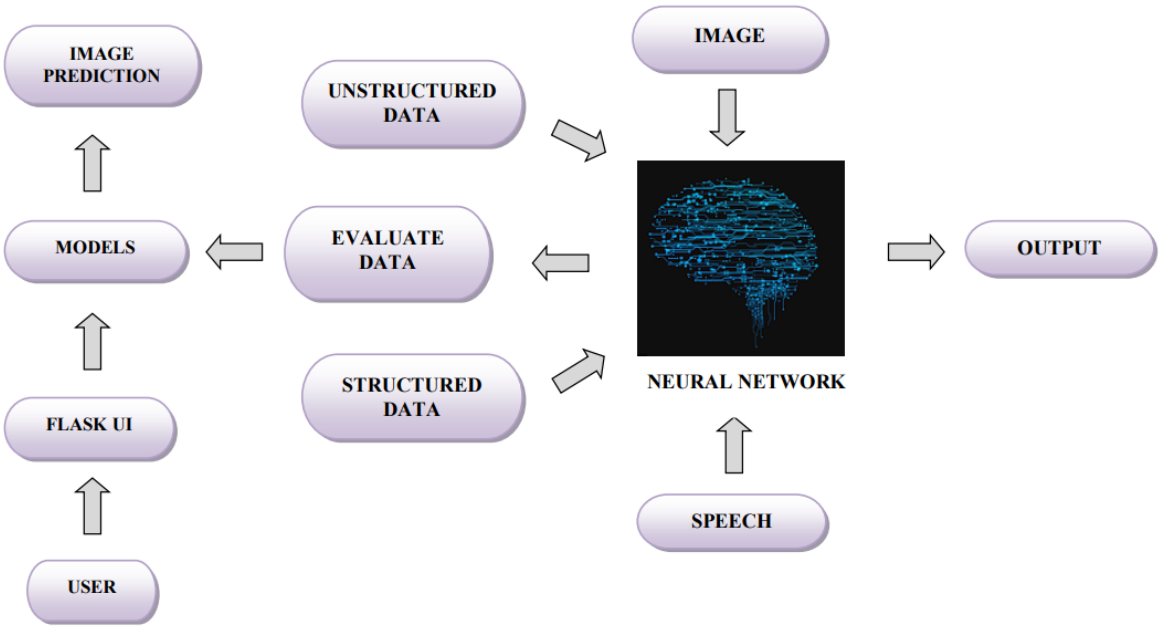


5.2 Solution and Technical Architecture

1. Technical Architecture
2. Solution Architecture



Solution Architecture



6. PROJECT PLANNING & SCHEDULING

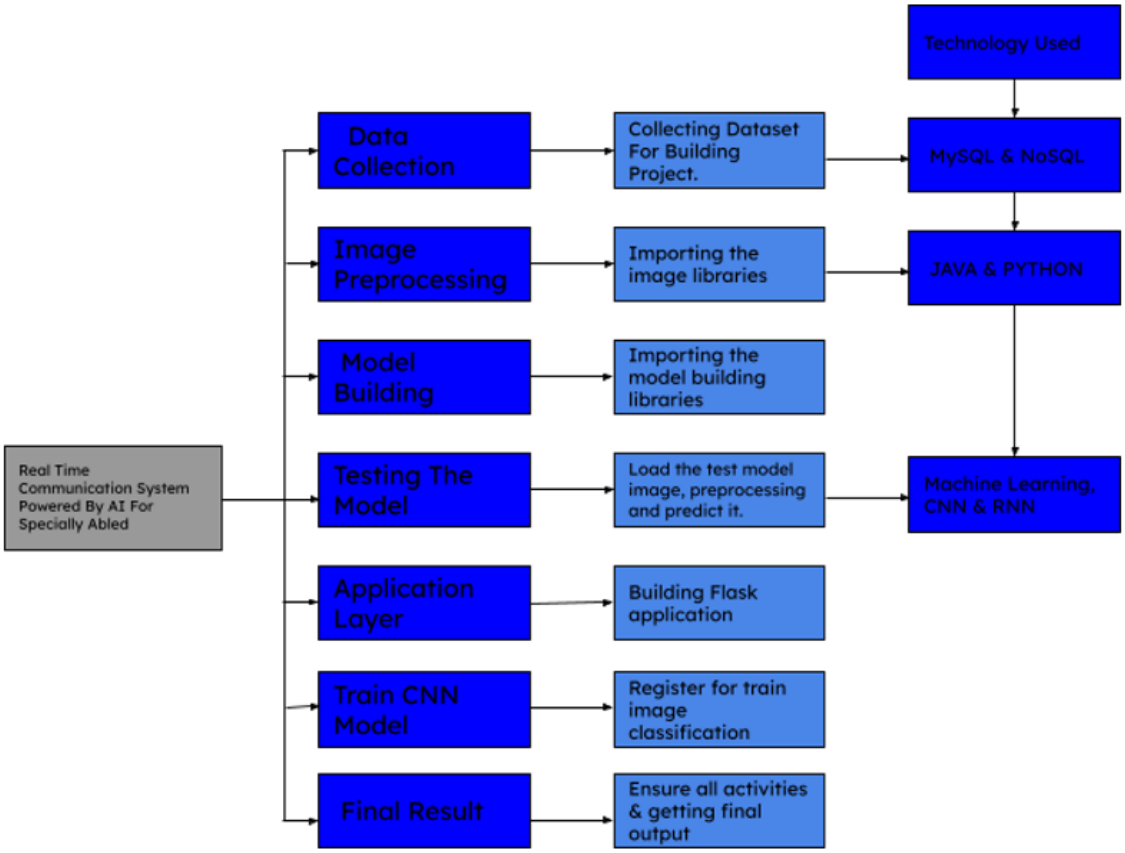
Use the below template to create product backlog and sprint schedule

SPRINT	FUNCTIONAL REQUIREMENT [EPIC]	USER STORY NUMBER	USER STORY / TASK	STORY POINTS	PRIORITY	TEAM MEMBER
Sprint-1	Data Collection	USN-1	Collect Dataset.	9	High	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-1		USN-2	Image preprocessing	8	Mediun	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	10	High	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-2		USN-4	Training the image classification model using CNN	7	Medium	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-3	Training And Testing	USN-5	Training the model and testing the model's performance	9	High	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-4	Implementation of the application	USN-6	Converting the input sign language images into English alphabets	8	Medium	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar

Milestone Activity Plan

MILESTONE	FUNCTION	MILESTONE STORY NUMBER	STORY / TASK
Milestone 1	Data collection	M1	We're collecting dataset for building our project and creating two folders, one for training and another one for testing.
Milestone 2	Image preprocessing	M2	Importing image data generator libraries and applying image data generator functionality to train the test set.
Milestone 3	Model building	M3	Importing the model building libraries, Initializing the model, Adding Convolution layers, Adding the Pooling layers, Adding the Flatten layers, Adding Dense layers, Compiling the model Fit and Save the model.
Milestone 4	Testing the model	M4	Import the packages first. Then we save the model and Load the test image, preprocess it and predict it.
Milestone 5	Application layer	M5	Build the flask application and the HTML pages.
Milestone 6	Train CNN model	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 list



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

SPRINT	FUNCTIONAL REQUIREMENT [EPIC]	USER STORY NUMBER	USER STORY / TASK	STORY POINTS	PRIORITY	TEAM MEMBER
Sprint-1	Data Collection	USN-1	Collect Dataset.	9	High	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-1		USN-2	Image preprocessing	8	Mediun	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	10	High	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-2		USN-4	Training the image classification model using CNN	7	Medium	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-3	Training And Testing	USN-5	Training the model and testing the model's performance	9	High	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar
Sprint-4	Implementation of the application	USN-6	Converting the input sign language images into English alphabets	8	Medium	1.GowthamKannan 2.Chithrai Selvan 3.Kanikhan Choudry 4.Arun Kumar

6.2 Sprint Delivery Schedule

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE [PLANNING]	STORY COMPLETED [AS ON PLANNED END DATE]	SPRINT RELEASE DATE [ACTUAL]
Sprint - 1	10	6 Days	24 Oct 2022	29 Oct 2022	8	18 Nov 2022
Sprint - 2	10	6 Days	31 Oct 2022	04 Nov 2022	5	18 Nov 2022
Sprint - 3	10	6 Days	07 Nov 2022	11 Nov 2022	7	18 Nov 2022
Sprint - 4	10	6 Days	14 Nov 2022	18 Nov 2022	5	18 Nov 2022

VELOCITY:

AV = Sprint Duration / Velocity

AV = 6/10

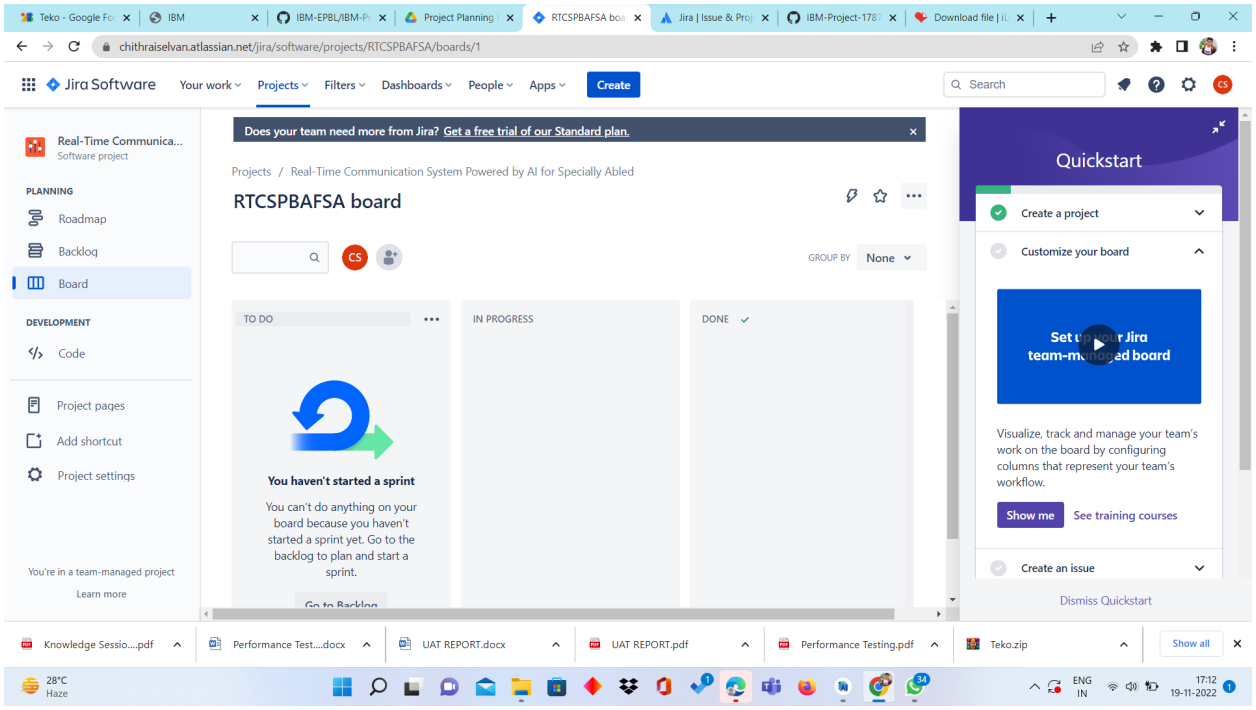
= 0.6

BURNDOWN CHART :

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

SPRINT BURNDOWN CHART :

6.2 REPORTS FROM JIRA



7. CODING AND SOLUTIONING

7.1 Libraries to be installed

pip install fer pip install flask

pip install cv2

pip install numpy

pip install keras

pip install tensorflow

pip install cvzone

pip install pyttsx3

pip install scikit-image

7.2 Real time sign to speech

Sign language is generally used by the people who are unable to speak, for communication. Most people will not be able to understand the Universal Sign

Language (unless they have learnt it) and due to this lack of knowledge about the language, it is very difficult for them to communicate with mute people. A device that helps to bridge a gap between mute persons and other people forms the crux of this project. Our system makes use of a model build using CNN that is capable of detection sign languages real time.

7.3 Facial Emotion Detection

Our system makes use of the FER model. Facial Emotion Recognition (commonly known as FER) is one of the most researched fields of computer vision till date and is still in continuous evaluation and improvement. The model is a convolutional neural network with weights saved to HDF5 file in the data folder relative to the module's path. It can be overridden by injecting it into the FER() constructor during instantiation with the emotion_model parameter.

7.4 Language Customization

Google Translate is a free multilingual machine translation service. It can translate the Website's text content from one language to another. It offers a huge list of languages to translate and has an efficient, reliable and easy way to translate the webpage in whatever language the user wants. It supports over 100 languages. Use this website translator to convert webpages into your choice of language.

7.5 Real time speech to text

With the Web Speech API, we can recognize speech using JavaScript. It is super easy to recognize speech in a browser using JavaScript and then get the text from the speech to use as user input. We use the Speech Recognition object to convert the speech into text and then display the text on the screen. Our system is capable of doing this over real-time. It is capable of recognizing any language in which the user is trying to communicate. But the support for this API is limited to the Chrome browser only. So if you are viewing this example in some other browser, the live example below might not work.

8. Testing

8.1 Test Cases

- Verify if the user can see the options when user clicks the URL
- Verify if the UI elements are getting displayed properly
- Verify if the user can choose any languages
- Verify if the user is getting redirected to the sign-to-speech page
- Verify if the application can convert the sign to speech
- Verify if the user can exit the sign-to-speech page
- Verify if the user is getting redirected to the speech-to-sign page
- Verify if the UI elements are being displayed
- Verify if the application can convert speech to text by clicking the voice to text button.
- Verify if the user can exit the speech-to-sign page

8.2 UAT Testing .

1. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

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

2. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

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

3. Performance Testing

Locust Test Report

During: 11/24/2022, 10:48:17 AM - 11/24/2022, 10:49:19 AM

Target Host: http://127.0.0.1:5000

Script: locustfile.py

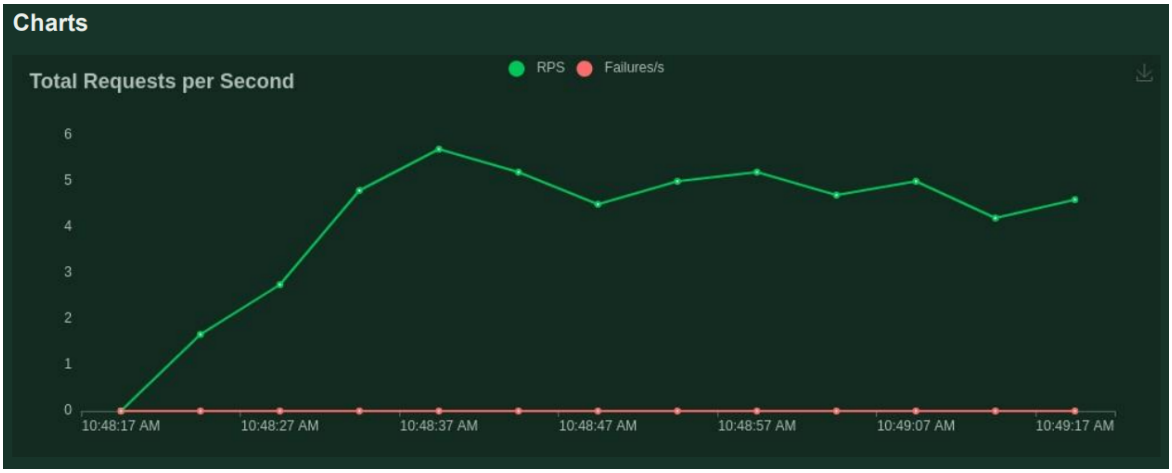
Request Statistics

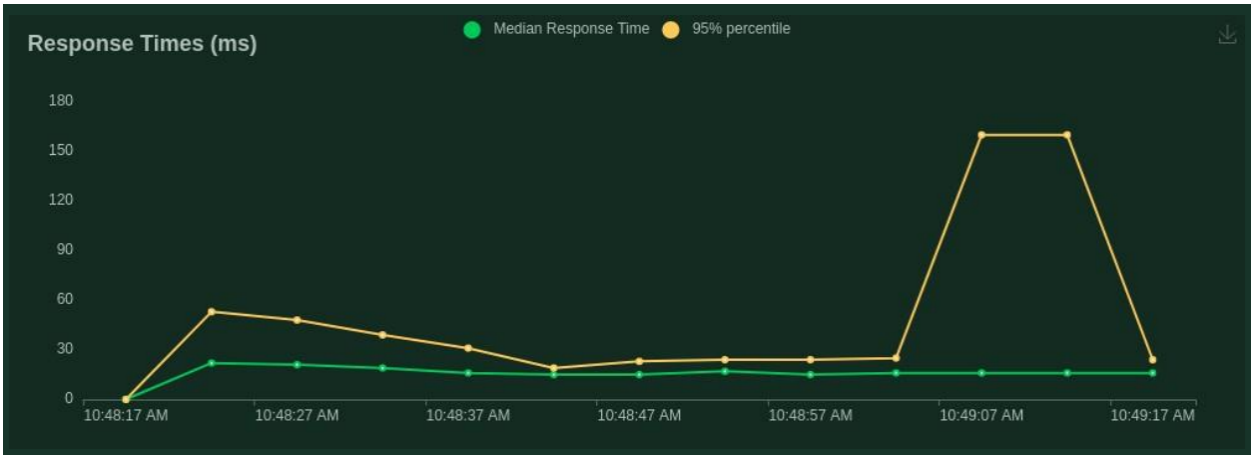
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	298	0	19	4	178	6317	4.8	0.0
	Aggregated	**298**	**0**	**19**	**4**	**178**	**6317**	**4.8**	**0.0**

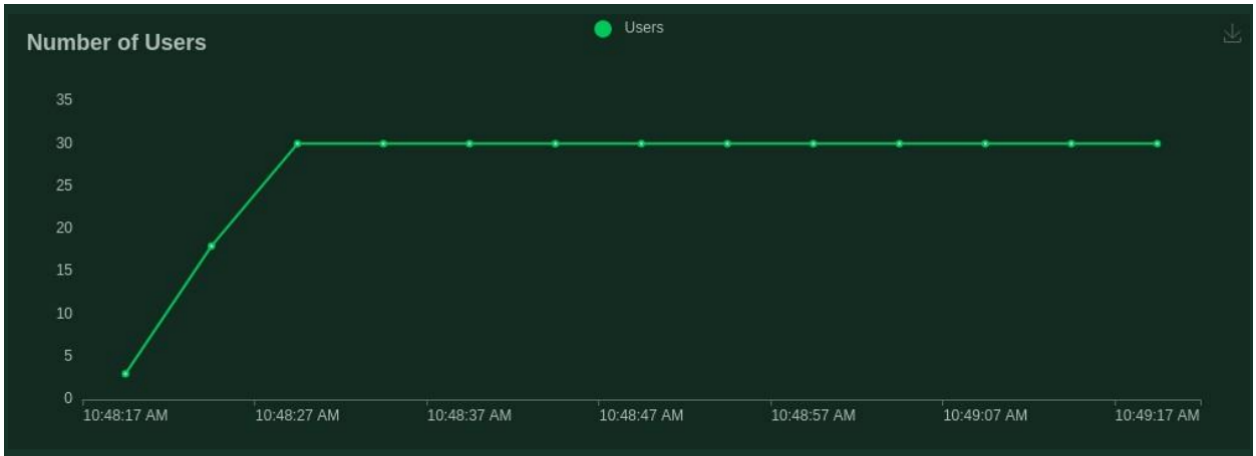
Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	16	17	19	22	25	32	80	180
	Aggregated	**16**	**17**	**19**	**22**	**25**	**32**	**80**	**180**

Charts







Final ratio

Ratio per User class

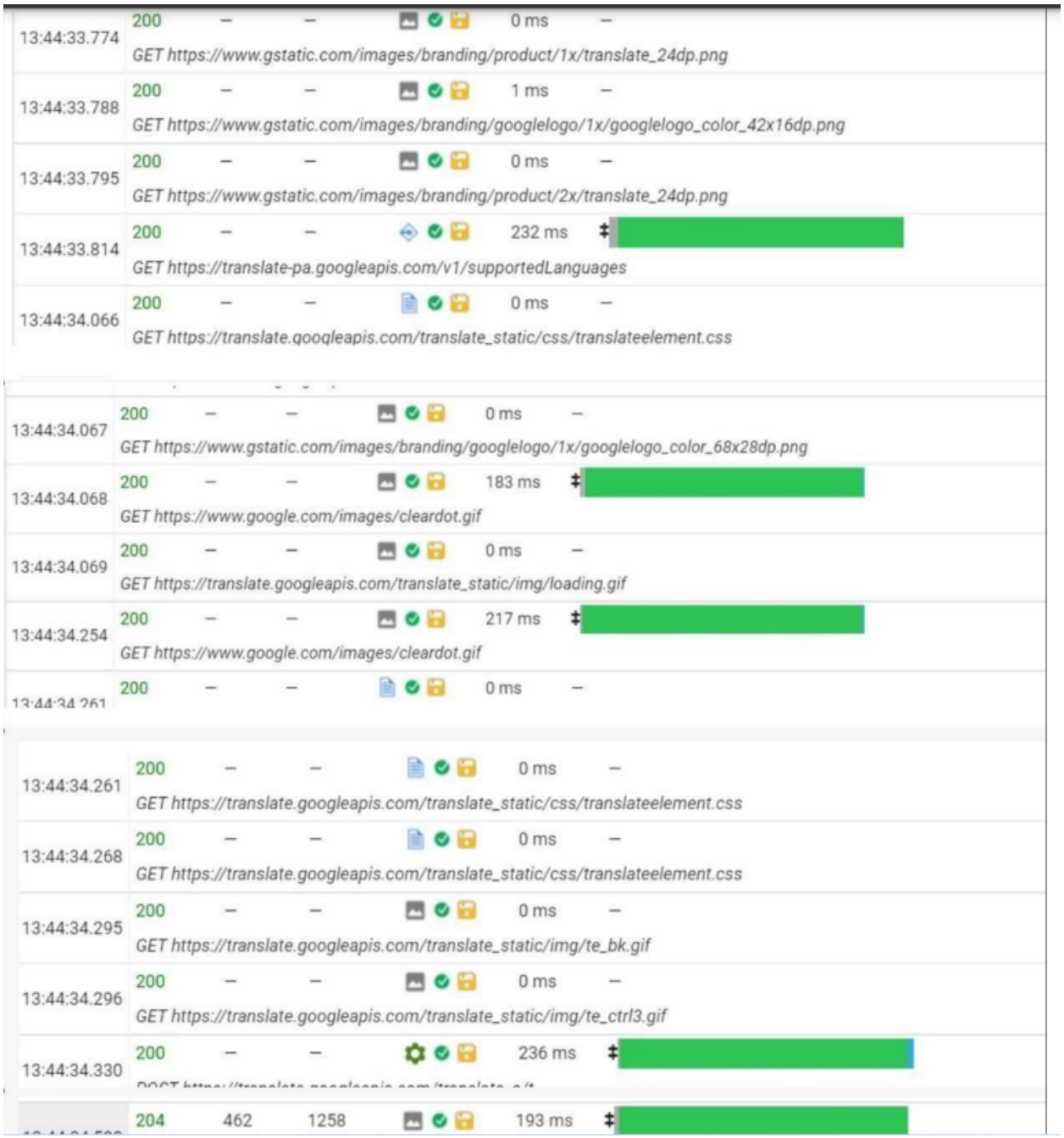
- 100.0% AppUser
  - 50.0% video_feed
  - 50.0% index

Total ratio

- 100.0% AppUser
  - 50.0% video_feed
  - 50.0% index

9. Results Performance Metrics

The following images can be studied to understand the performance metrics of our system



10. Advantages and Disadvantages

Advantages:

- Real-time sign-to-speech detection.
- Model provides good accuracy.
- Real-time facial emotion detection.
- Language Customization.
- Real time speech-to-text conversion.
- Friendly UI
- Data privacy

Disadvantages:

- At times the website may lag.
- Model is not tested on a wide set of data set, having all the signs.
- Sign language customization feature is not available.
- User cannot take notes while using the app.
- User cannot make calls using the app.
- Speech recognition works only on google chrome.

11.Conclusion

Communication is crucial for self-expression. Additionally, it meets one's necessities. Effective communication is necessary for career advancement. Effective communication skills can make your personal life easier and improve your interactions with others by facilitating mutual understanding. A system that translates speech into acceptable sign language for the deaf and dumb has been developed as part of our project. It also translates sign language into a human hearing voice to communicate with average people. A convolution neural network has been used to build a model that is trained on various hand motions. Utilizing this concept, an app is created. Through the use of signs that are translated into speech and human-understandable English, this software aids deaf and dumb individuals to communicate easily**.**

12. Future Scope

The following are the features that can be added to our application:

- ➔ A communication app can be built with the same set of features. The user can choose the appropriate mode (speech to sign or sign to speech) and accordingly the real-time detection would take place on both the end user's applications.
- ➔ The accuracy of the model shall be increased.
- ➔ Customization of languages shall be added
- ➔ Users shall be allowed to write notes while on call.
- ➔ Customization of signs can also be added as a feature.

13. Appendix Source Code

13.1 Source Code

1.DATA COLLECTION.py

```
import cv2

from cvzone.HandTrackingModule import HandDetector

import numpy as np

import math

import time


cap = cv2.VideoCapture(0)

detector = HandDetector(maxHands=1)

offset = 20
```

```
imgSize = 300

folder = "../Data/1"
counter = 0
while True:
    success, img = cap.read()
    hands, img = detector.findHands(img)
    if hands:
        hand = hands[0]
        x, y, w, h = hand['bbox']

        imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
        imgCrop = img[y - offset:y + h + offset, x - offset:x + w + offset]

        imgCropShape = imgCrop.shape

        aspectRatio = h / w

        if aspectRatio > 1:
            k = imgSize / h
```

```
            wCal = math.ceil(k * w)
            imgResize = cv2.resize(imgCrop, (wCal, imgSize))
            imgResizeShape = imgResize.shape
            wGap = math.ceil((imgSize - wCal) / 2)
            imgWhite[:, wGap:wCal + wGap] = imgResize
        else:
            k = imgSize / w
            hCal = math.ceil(k * h)
            imgResize = cv2.resize(imgCrop, (imgSize, hCal))
            imgResizeShape = imgResize.shape
            hGap = math.ceil((imgSize - hCal) / 2)
            imgWhite[hGap:hCal + hGap, :] = imgResize

        cv2.imshow("ImageCrop", imgCrop)
        cv2.imshow("ImageWhite", imgWhite)
        cv2.imshow("Image", img)
    key = cv2.waitKey(1)
    if key == ord("s"):
        counter += 1
        cv2.imwrite(f'{folder}/Image_{time.time()}.jpg', imgWhite)
```

```
    print(counter)
```

2.CAMERA.py

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


class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        # self.model = load_model('asl_model.h5') # Execute Local Trained Model
        self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
        # self.model = load_model('keras_model.h5') # Execute IBM Trained Model
        self.index=['A']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
```

```
            ret,frame = self.video.read()
            frame = cv2.resize(frame, (640, 480))
            copy = frame.copy()
            copy = copy[150:150+200,50:50+200]
            # Prediction Start
            cv2.imwrite('image.jpg',copy)
            copy_img = image.load_img('image.jpg', target_size=(64,64))
            x = image.img_to_array(copy_img)
            # pred_array_scaled = np.expand_dims(pred_array_scaled, axis=0)
            x = np.expand_dims(x, axis=0)
            pred = np.argmax(self.model.predict(x), axis=1)
            self.y = pred[0]
            cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
            ret,jpg = cv2.imencode('.jpg', frame)
            return jpg.tobytes()
```

3.APP.py

```
from flask import Flask, render_template,Response
from flask import Flask, Response, render_template
from camera import Video
app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')
def gen(camera):
    while True:
        frame = camera.get_frame()
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +b'\r\n\r\n')
@app.route('/video_feed')
def video_feed():
    video = Video()
    return
Response(gen(video), mimetype='multipart/x-mixed-replace; boundary
= frame')
if __name__ == '__main__':
    app.debug = True
```

```
    app.run()
```

4.MAIN.py

```
import cv2
video = cv2.VideoCapture(0)


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


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

5.INDEX.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/html">
<head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width,
initial-scale=1">
```

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

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.
min.css">

    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.12.0/css/all.css">

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




</head>

<body>



<header>

    <section class="header">

<span class="logo">



     <img src="{{ url_for('static', filename='img/job.png') }}"
width="100%">



</span>



        <span class="logoname">
```

Real-Time Communication System Powered by AI for Specially Abled

<nav>

<ul class="navbar">

 Home

 About

 Blog

 Contact

</nav>

</header>

<section>

<div class="d-flex flex-column justify-content-center align-items-center" id="div-video-feed"

style="width: 640px;height: 480px;margin: 10px;min-height: 480px;min-width: 640px;border-radius: 10px;border: 4px dashed rgb(255,255,255) ;">

```
            <img src="{{ url_for('video_feed') }}" style="width: 100%;height:
100%;color: rgb(255,255,255);text-align: center;font-size: 20px;"

                alt="Camera Access Not Provided camera!">

        </div>

    </div>

    <div class="d-flex flex-column justify-content-center
align-items-center" style="margin-bottom: 10px;"><button

            class="btn btn-info" type="button" data-bs-target="#modal-1"
data-bs-toggle="modal">Quick Reference

        -<strong> ASL Alphabets</strong></button></div>

</section>



<section>

    <div class="container">

        <div class="accordion text-white" role="tablist" id="accordion-1">

            <div class="accordion-item" style="background:
rgb(33,37,41);">

                <h2 class="accordion-header" role="tab"><button
class="accordion-button" data-bs-toggle="collapse"

                                                    data-bs-target="#accordion-1
.item-1" aria-expanded="true"

                                                    aria-controls="accordion-1
.item-1"

                                                    style="background:
rgb(39,43,48);color: rgb(255,255,255);">About The
Project</button></h2>
```

```
            <div class="accordion-collapse collapse show item-1"
role="tabpanel" data-bs-parent="#accordion-1">

                <div class="accordion-body">

                    <p class="mb-0">Artificial Intelligence has made it
possible to handle our daily activities

                        in new and simpler ways. With the ability to
automate tasks that normally require human

                        intelligence, such as speech and voice recognition,
visual perception, predictive text

                        functionality, decision-making, and a variety of other
tasks, AI can assist people with

                        disabilities by significantly improving their ability to
get around and participate in

                        daily activities.<br><br>Currently, Sign Recognition
is available <strong>only for

                            alphabets A-I</strong> and not for J-Z, since J-Z
alphabets also require Gesture

                        Recognition for them to be able to be predicted
correctly to a certain degree of

                        accuracy.</p>

                </div>

            </div>

        </div>

        <div class="accordion-item" style="background:
rgb(33,37,41);">

            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed"
```

```
                                                    data-bs-toggle="collapse"
data-bs-target="#accordion-1 .item-2" aria-expanded="false"

                                                    aria-controls="accordion-1
.item-2"

                                                    style="background:
rgb(39,43,48);color: rgb(231,241,255);">Developed By</button></h2>

            <div class="accordion-collapse collapse item-2"
role="tabpanel" data-bs-parent="#accordion-1">

                <div class="accordion-body">

                   <p class="mb-0">AGNI COLLEGE OF TECHNOLOGY

                      <br><br>1. <strong>Chithrai Selvan S      </strong>
312819104019

                      <br>2. <strong>Arun Kumar M              </strong>
312819104011

                      <br>3. <strong>Gowtham Kannan R          </strong>
312819104032

                      <br>4. <strong>Kannikhan Choudry D       </strong>
312819104039

                   </p>

                </div>

             </div>

          </div>

       </div>

   </div>

</section>
```

```
<div class="modal fade" role="dialog" tabindex="-1" id="modal-1">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h4 class="modal-title">American Sign Language - Alphabets</h4><button type="button"
class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <img src="{{ url_for('static', filename='img/sign-language.jpg') }}" width="100%">
            </div>
            <div class="modal-footer"><button class="btn btn-secondary" type="button"
data-bs-dismiss="modal">Close</button></div>
        </div>
    </div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
```

```
</body>
</html>
```

6.Style.css

```
@font-face{
    font-family: "DancingScript";
    src: url("../fonts/DancingScript-VariableFont_wght.ttf");
}
@font-face{
    font-family: koulen;
    src: url("../fonts/Koulen-Regular.ttf");
}
@font-face{
    font-family: Dongle;
    src: url("../fonts/Dongle-Regular.ttf");
}
@font-face{
    font-family: babas;
    src: url("../fonts/BebasNeue-Regular.ttf");
}
*{
    margin: 0px;
    padding: 0px;
```

```
}

.fit-cover {
  object-fit: cover;
}
body{
        background-image: url("../images/background.jpg");
    background-repeat: no-repeat;
        background-position: cover;



}

.logo{
        width: 50px;
}

.header{
        width: 100%;
        display: flex;
        align-items: center;
        justify-content: space-between;
```

```
        background-color: #97C4B8;
        padding: 10px 10px;
        position: sticky;
        box-shadow: 0px 2px 10px black;
}

.logoname{
        font-size: 40px;
        position: absolute;
        margin-left: 80px;
        font-family: DancingScript;
}

.navbar{
        display: flex;
}

.navbar li{
        font-size: 20px;
    list-style: none;
    padding: 0px 20px;
}
```

```
.navbar li a{
        text-decoration: none;
        color: #F9F3EE;
        font-family: koulen;
        transition: 0.2s ease;
}

.navbar li a:hover{
        color: green;

}

.active{
        color: black !important;
        text-decoration: underline !important;
}


.login{
        background-color: white;
        position: absolute;
```

```
        margin-left: 10%;
        margin-top: 10%;
        border-radius: 15px;
        padding: 25px 25px;
        border-style:double;
/*
        border-style: solid;
        border-color: black;
        border-width: -2px;
*/
}

.login:hover{
        box-shadow: 2px 2px 10px black;

}
.log {
        display: flex;
        flex-direction: column;
        text-align: center;
        font-family: 'Koulen';
        font-size: 30px;
```

```
        padding-top: 1px;
        margin: 0;

}

.btnline{
        display: flex;
        margin: 0px 30px;
        width: 77%;
        border-bottom: 1px solid silver;
}
.login form {
        padding: 0 40px;
        box-sizing: border-box;
        padding: 10px 30px;

}

form .fieldname{
        position: relative;
        border-bottom: 2px solid #adadad;
        width: 100%;
```

```
        font-family: dongle;



}
.fieldname input{
        width: 100%;
        padding: 0 5px;
        height: 40px;
        font-size: 16px;
        background: none;
        outline: none;
        border: none;


}
.fieldname label{
        position: absolute;
        top: 50%;
        left: 5px;
        color: #adadad;
        transform: translateY(-50%);
```

```
        font-size: 25px;
        pointer-events: none;
        transition: .3s;
}
.fieldname span::before {
    content: '';
    position: absolute;
    top: 40px;
    left: 0;
    width: 0%;
    height: 2px;
    background: #2691d9;
    transform: .5s;
}

.fieldname input:focus ~ label,
.fieldname input:valid ~ label{
        top: 5px;
        color: #2691d9;
}

.fieldname input:focus ~ span::before,
```

```
.fieldname input:valid ~ span::before{
        width: 100%;
}



.pass{
        margin:-5px 0 20px 5px;
        cursor: pointer;
        text-align: center;
}
.pass a {
        color: #adadad;
        text-decoration: none;
        font-family: babas;
        font-size: 15px;

}
.pass:hover {
        text-decoration: underline;
}

input[type="submit"]{
```

```
        width: 100%;
        height: 30px;
    border: 2px;
        background: #97C4B8;
        border-radius: 25px;
        font-size:20px;
        color: white;
        font-weight: 70;
        cursor: pointer;
        outline: none;
        font-family: babas;

}
input[type="submit"]:hover{

        border: 2px solid #68A7AD;

}

.signup{
        text-align: center;
        padding: 20px;
```

```
        font-family: babas;
}
.signup a{
        text-decoration: none;
        font-family: babas;
}
.signup a:hover{
        color: Green;
}

.search_form{
        padding: 10px 10px;
        background: none;

}

.search_form input[type="text1"]{
        position: relative;
        background-color: none;
        outline: none;
        border-radius: 25px;
        width: 15rem;
```

```
        height: 2rem;
        padding: 2px 10px;
        font-family: poppins, monospace;
        letter-spacing: 1px;
        left: 82%;
        border: 2px solid #adadad;
        cursor: pointer;
        transition: .3s;

}

form input[type="text1"]:hover{
        border: 2px solid black;
}

#searchicon{
        position: absolute;
        margin-left: 95.5%;
        margin-top: .5rem;
        z-index: 1;
        cursor: pointer;
        color: #adadad;
```

```
        transition: .3s;
}
#searchicon:hover {
        color: black;

}

.categorytitle h1{
        font-family: DancingScript;
        text-align: center;
        padding-top: 5px 0px;
        font-size: 50px;
        background-color: purple;
        border-radius: 90px 90px 0px 0px ;
        position: relative;
        box-shadow: 0px 1px 5px black;
}
.category{
        display: flex;
        justify-content: space-between;
    padding-top: 1px;
        padding-bottom: 1px;
```

```
        background-color: #97C4B8;
        width: 100%;
        box-shadow: 0px 2px 10px black;

}

.category a{
        text-decoration: none;
        /*color: #AB46D2;*/
        font-size: 20px;
}

.category li{
        margin-left: 70px;
        margin-right: 70px;
        list-style: none;
        border-radius: 10px;
        padding: 5px 15px;
        font-family: koulen, 'monospace';
        justify-content: space-between;
}
```

```
.category a:hover{
        color: #4D77FF;

}
.layer{
        background-color: darkred;
    position: relative;
        margin: 20px 0px 30px 20px;
        padding: 20px;
}

.imglayer{
                background-color: pink;
                width: 20%;
                padding: 20px;
}
.imglayer img{
        width: 100%;
}
.decriptionlayer{
        background-color: yellow;
}
```

13.2 GitHub & Project Demo Link

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-35661-1660287273.git