# Final Report

# A Real-Time Communication System For Specially Abled


## Team Members:

1. Gowtham kannan R
2. Chithrai selvan
3. Kannikhan choudry
4. Arun kumar M

# Table Of Contents

# 1. Introduction

## 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

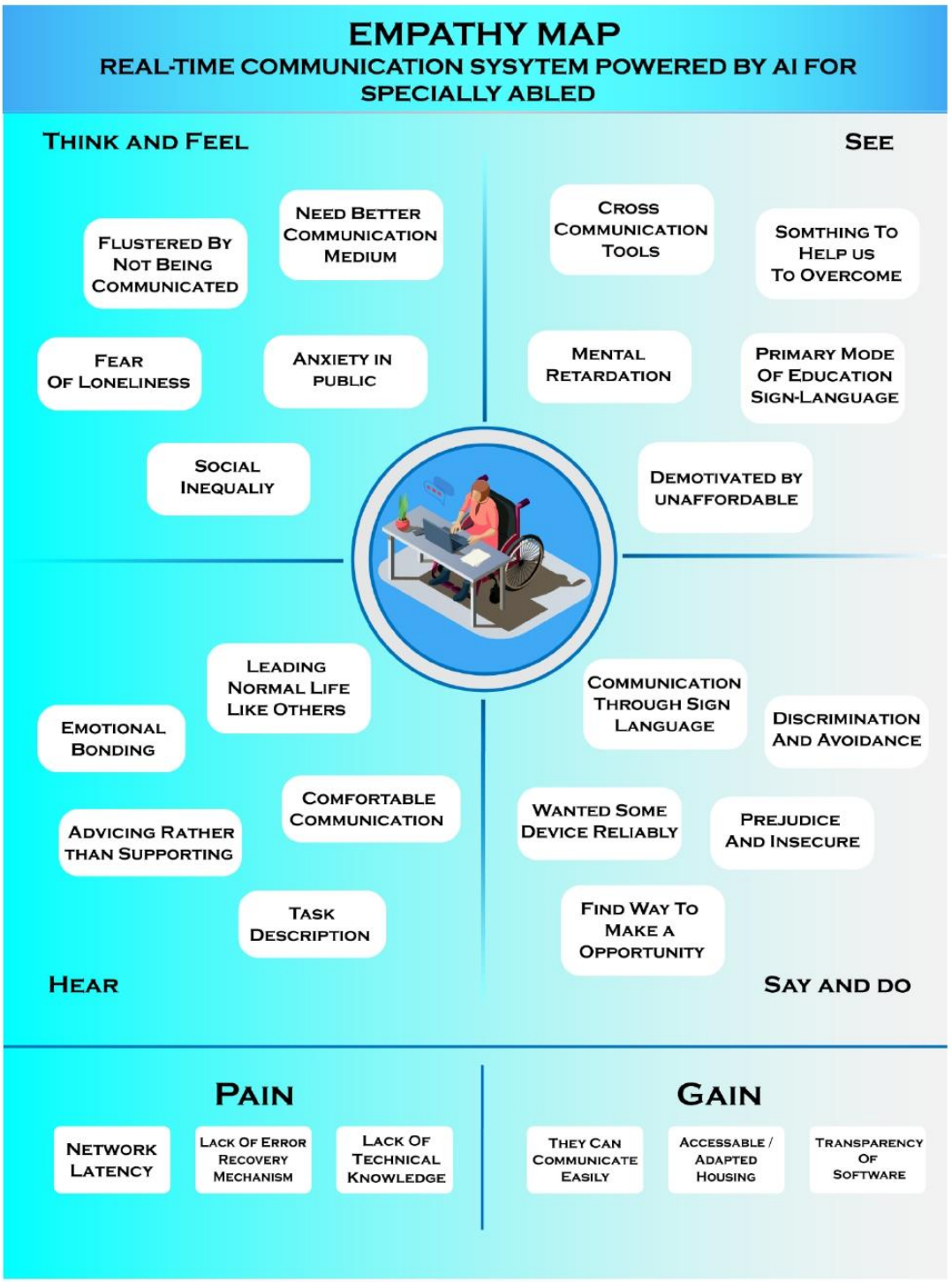
deep neural network (DNN) to map from acoustic features to the time-domain glottal flow waveform. First, acous-tic features and the glottal flow signal are estimated from each frame of the speech database. Pitch-synchronous glottal flow time-domain waveforms are extracted, interpolated to a constant 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

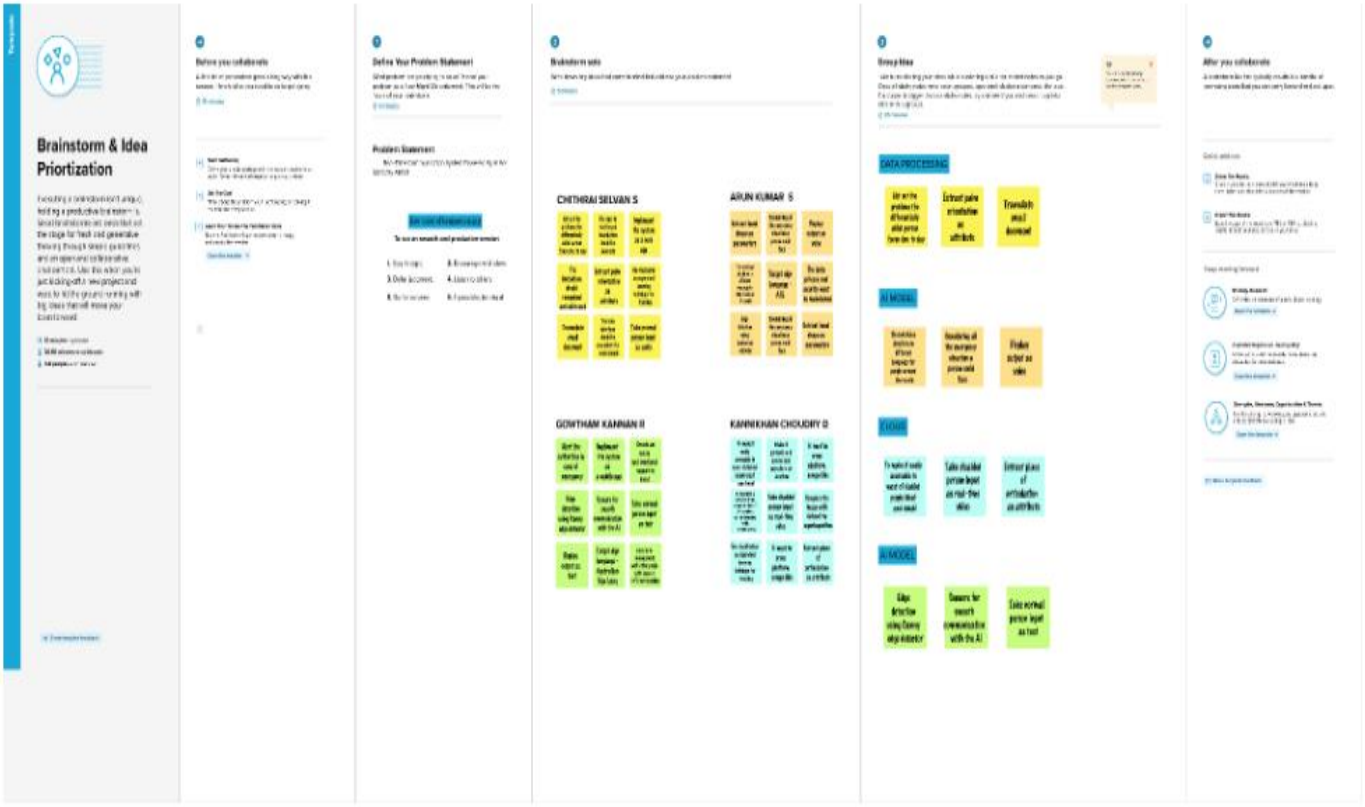
People with disabilities are a part of our society. Even though technology is constantly evolving, little is being done to improve the lives of these people. Communication with a deaf-mute person has always been difficult. Because hand sign language is not taught to the general public, it can be difficult 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.

# 3. Ideation and Proposed Solution

## 3.1 Empathy Map Canvas



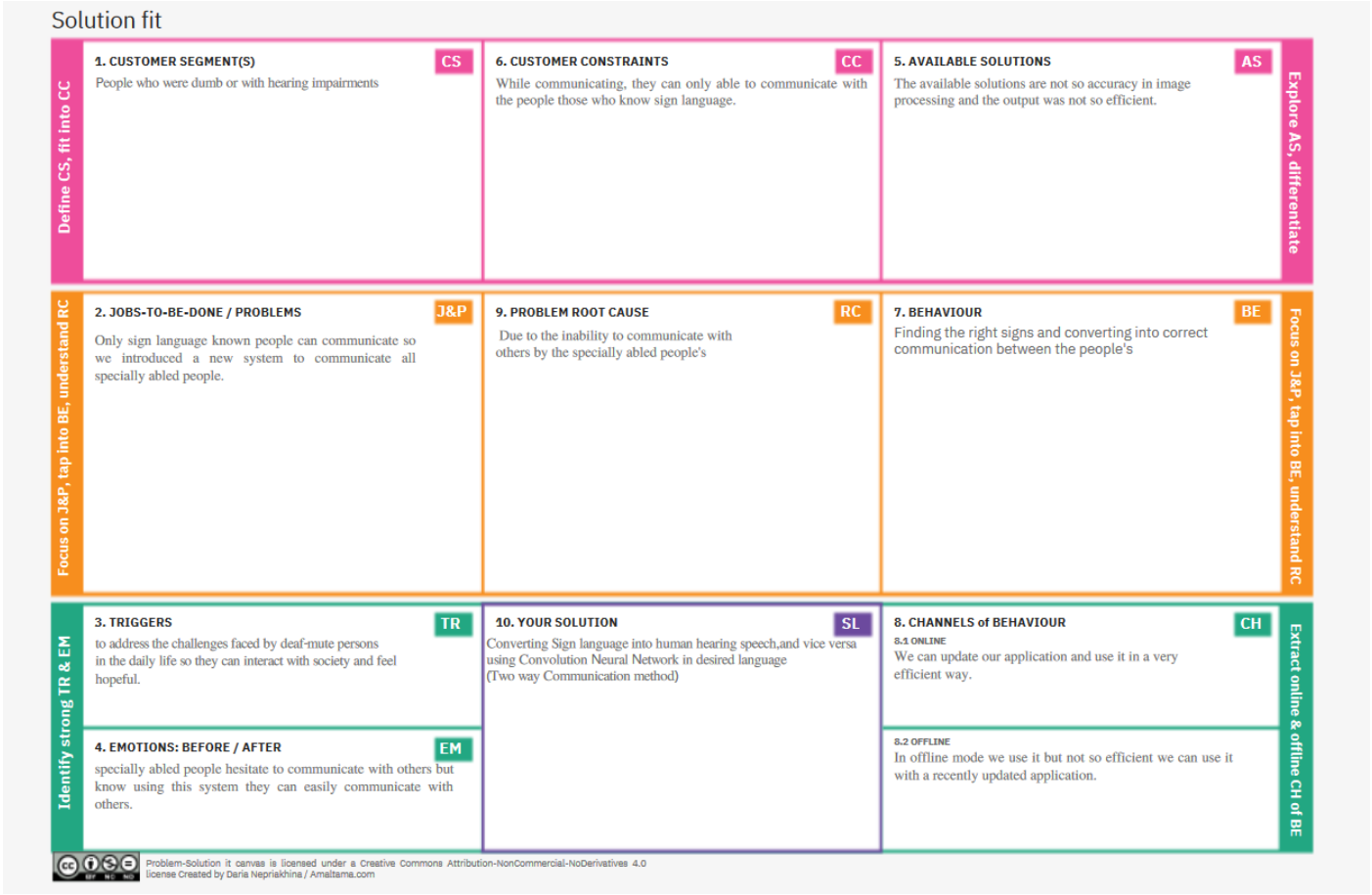
# 3.2 Ideation and Brainstorming



## 3.3 Proposed Solution

| 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

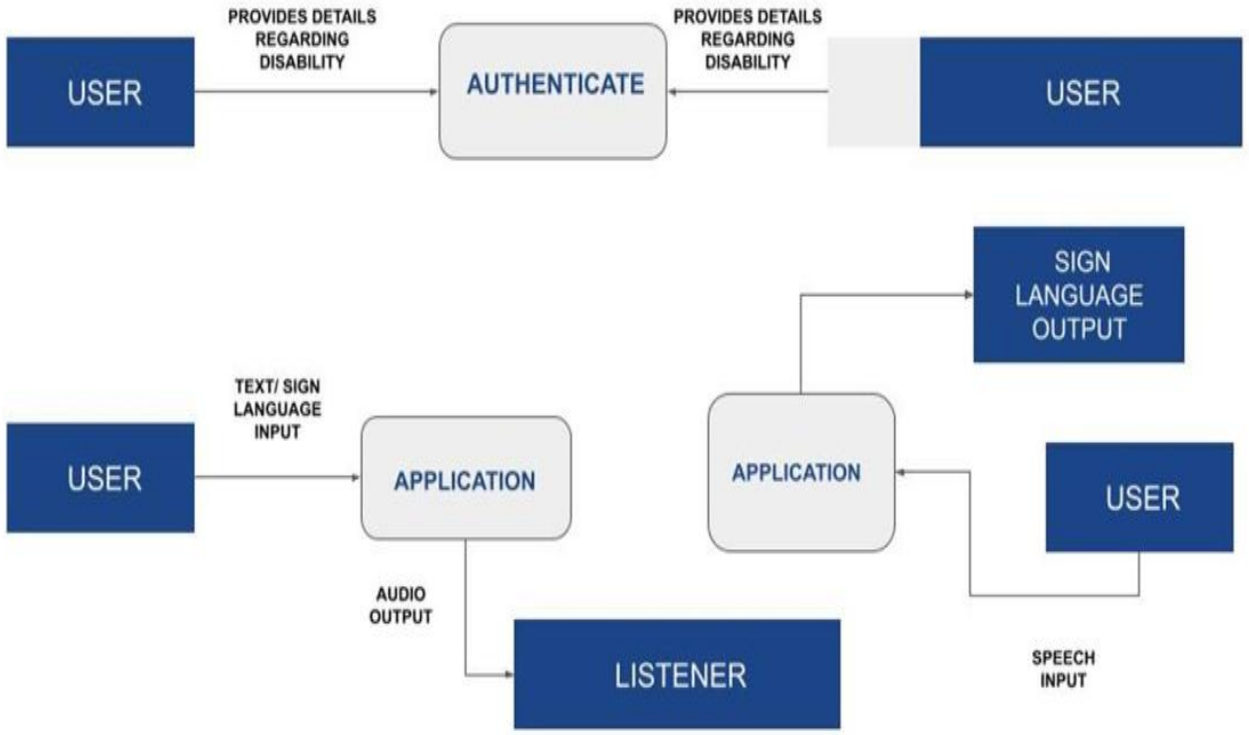
| FR No. | Functional Requirement (Epic) | Sub Requirement (Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration is done through Gmail |
| FR-2 | User Confirmation | Confirmation via Email |
| FR-3 | Communication requirement | For one on one mentoring, teacher will be available. |
| FR-4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

## 4.2 Non-Functional Requirement

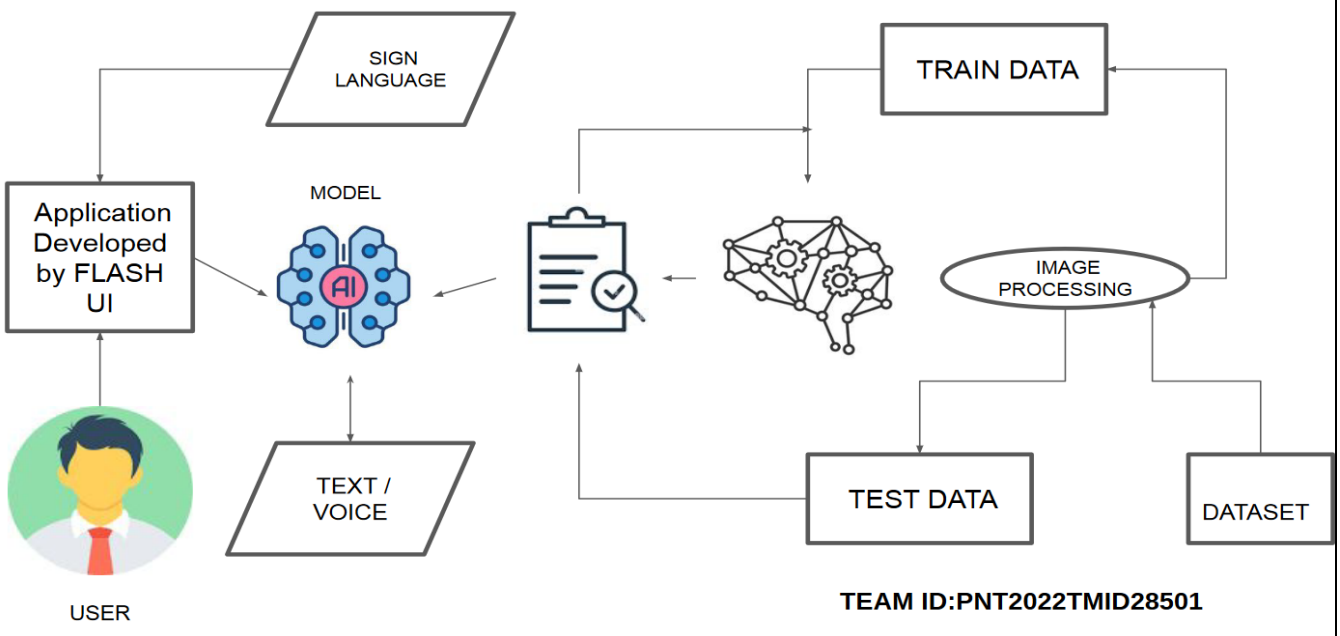
| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| **NFR-1** | **Usability** | The camera captures all expressions including facial expressions and hand gestures which can be easily used by all age groups. It can be used by deaf-mute people and their care takers. |
| NFR-2 | **Security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

# 5. Project Design

## 5.1 DATA FLOW DIAGRAM:

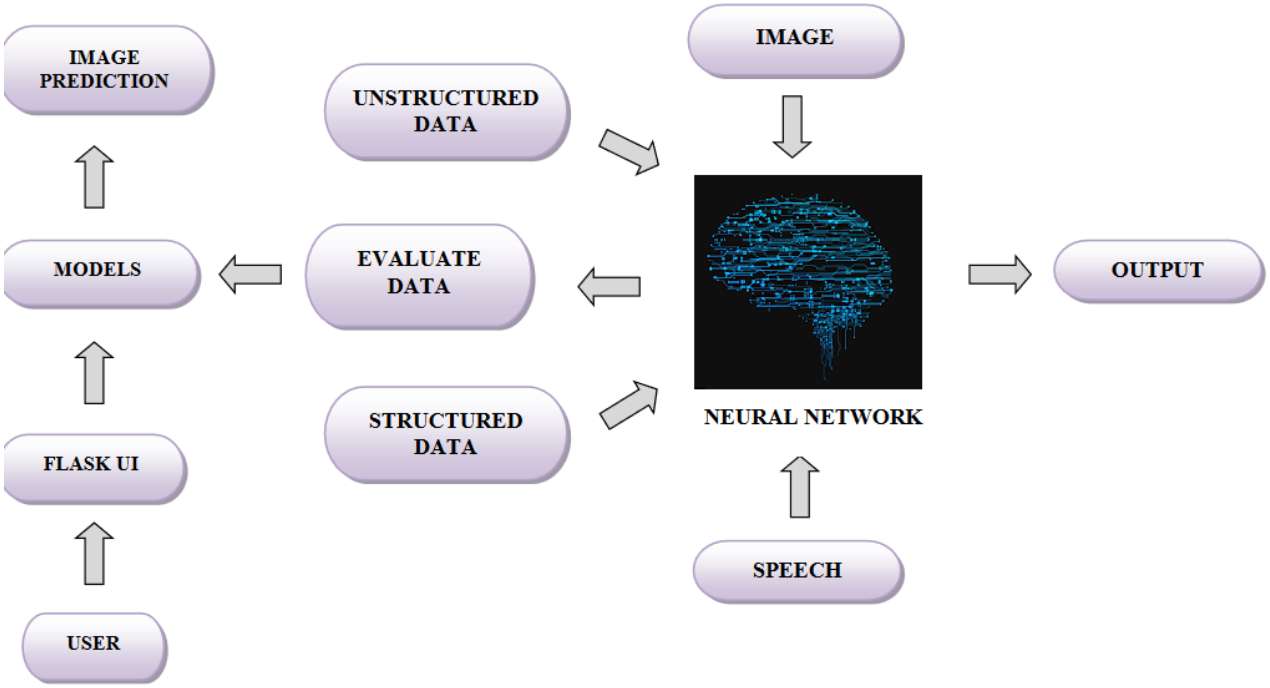



5.2 Solution and Technical Architecture




**TEAM ID:PNT2022TMID28501**

# Solution Architecture



| | **Functional Requirement (Epic)** | **User Story Number** | | **criteria** | | |
|---|---|---|---|---|---|---|
| Customer | Uploading the real time data. | USN-1 | The user will be presented with two options. Speech to sign language conversion. 2. Sign language to speech conversion. | They can access the portal | High | Sprint-1 |
| | | USN-2 | Language selection | They can access the portal | Low | Sprint-1 |
| | | USN-3 | The deaf-mute person will choose the speech to sign language conversion which would take them into a portal that collects the real time data (sign language recognition) and converts it into speech simultaneously. | Video processing | High | Sprint-2 |
| | | USN-4 | Emotion detection | Video processing | Medium | Sprint-1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | USN-5 | Normal person would choose speech to sign language which would take them into a portal where their speech is converted into sign language simultaneously. | Video and audio processing | High | Sprint-1 |

| User Story / Task | Acceptance | Priority | Release |
| --- | --- | --- | --- |

5.3 User Story **r**

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Data Collection | USN-1 | Collect Dataset. | 9 | High | G KAVIYA , USHA KUMARI |
| Sprint-1 | | USN-2 | Image pre-processing | 8 | Medium | B MOHAMED FAHAD, J VAISHNAVI |
| Sprint-2 | Model Building | USN-3 | Import the required libraries, add the necessary layers, and compile the model | 10 | High | G KAVIYA , USHA KUMARI |
| Sprint-2 | | USN-4 | Training the image classification model using CNN | 7 | Medium | B MOHAMED FAHAD, J VAISHNAVI |
| Sprint-3 | Training and Testing | USN-5 | Training the model and testing the model's performance | 9 | High | G KAVIYA , USHA KUMARI |
| Sprint-4 | Implementation of the application | USN-6 | Converting the input sign language images into English alphabets | 8 | Medium | B MOHAMED FAHAD, J VAISHNAVI. |

# 6.1 Sprint Planning and Estimation

**Product Backlog, Sprint Schedule, and Estimation**

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Data Collection | USN-1 | Collect Dataset. | 9 | High | G KAVIYA , USHA KUMARI |
| Sprint-1 | | USN-2 | Image pre-processing | 8 | Medium | B MOHAMED FAHAD, J VAISHNAVI |
| Sprint-2 | Model Building | USN-3 | Import the required libraries, add the necessary layers, and compile the model | 10 | High | G KAVIYA , USHA KUMARI |
| Sprint-2 | | USN-4 | Training the image classification model using CNN | 7 | Medium | B MOHAMED FAHAD, J VAISHNAVI |
| Sprint-3 | Training and Testing | USN-5 | Training the model and testing the model's performance | 9 | High | G KAVIYA , USHA KUMARI |
| Sprint-4 | Implementation of the application | USN-6 | Converting the input sign language images into English alphabets | 8 | Medium | B MOHAMED FAHAD, J VAISHNAVI. |

## 6.2 Sprint Delivery Schedule

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---|---|---|---|---|---|---|
| Sprint-1 | 10 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 8 | 29 Oct 2022 |
| Sprint-2 | 10 | 6 Days | 31 Oct 2022 | 04 Nov 2022 | 5 | 04 Nov 2022 |
| Sprint-3 | 10 | 6 Days | 07 Nov 2022 | 11 Nov 2022 | 7 | 11 Nov 2022 |
| Sprint-4 | 10 | 6 Days | 14 Nov 2022 | 18 Nov 2022 | 5 | 18 Nov 2022 |

**Project Tracker, Velocity & Burndown Chart:**

$$AV = \frac{sprint\ duration}{velocity}$$

**AV = 6/10 = 0.6**

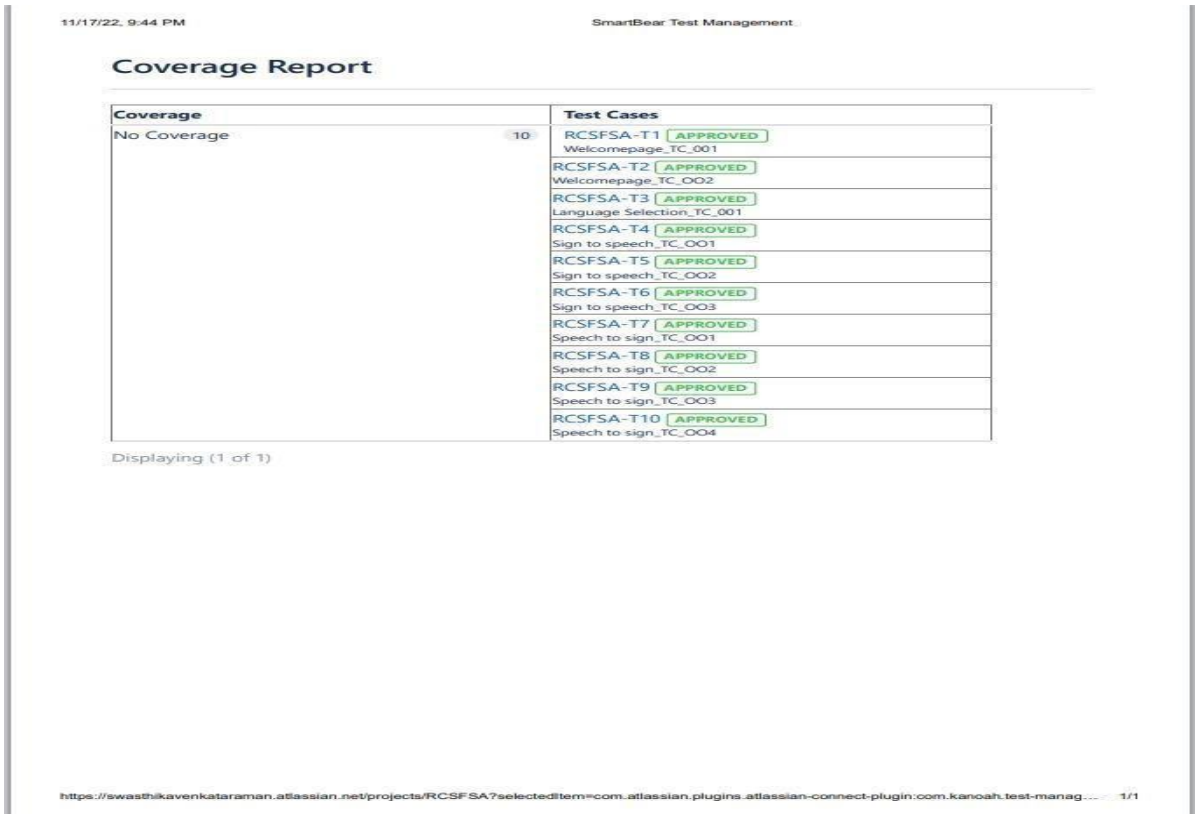
**Velocity:**


Imagine we have a 10-day sprint duration, and the velocity of the team is 6 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

**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

# 6.2 Reports From JIRA



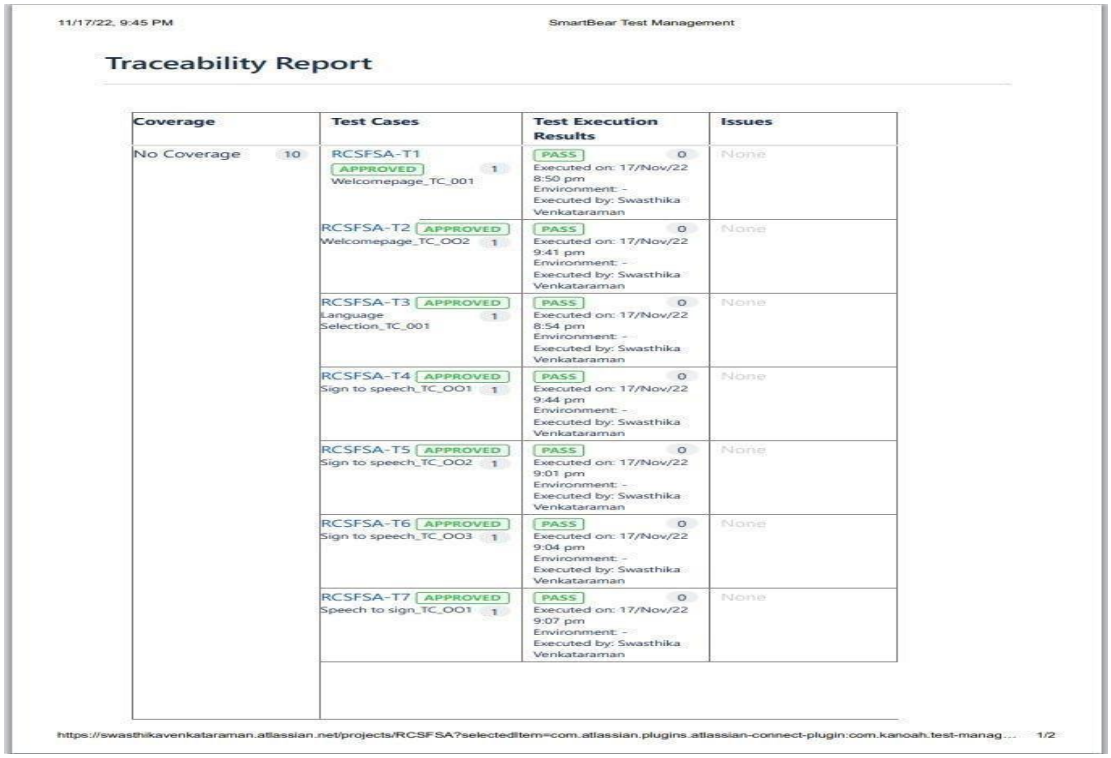
11/17/22, 9:44 PM SmartBear Test Management

## Coverage Report

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

Displaying (1 of 1)

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



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

## Traceability Report

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

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

| Coverage | Test Cases | Test Execution Results | Issues |
|---|---|---|---|
| | RCSFSA-T8 APPROVED<br>Speech to sign_TC_OO2 1 | PASS 0<br>Executed on: 17/Nov/22 9:12 pm<br>Environment: -<br>Executed by: Swasthika Venkataraman | None |
| | RCSFSA-T9 APPROVED<br>Speech to sign_TC_OO3 1 | PASS 0<br>Executed on: 17/Nov/22 9:18 pm<br>Environment: -<br>Executed by: Swasthika Venkataraman | None |
| | RCSFSA-T10 APPROVED 1<br>Speech to sign_TC_OO4 | PASS 0<br>Executed on: 17/Nov/22 9:24 pm<br>Environment: -<br>Executed by: Swasthika Venkataraman | None |

Displaying (1 of 1)



## Traceability matrix

| Coverage \ Test Cases | RCSFSA-T1 - Welcomepage_... | RCSFSA-T2 - Welcomepage_... | RCSFSA-T3 - Language Selec... | RCSFSA-T4 - Sign to speech_... | RCSFSA-T5 - Sign to speech_... | RCSFSA-T6 - Sign to speech_... | RCSFSA-T7 - Speech to sign_... | RCSFSA-T8 - Speech to sign_... | RCSFSA-T9 - Speech to sign_... | RCSFSA-T10 - Speech to sign... |
|---|---|---|---|---|---|---|---|---|---|---|
| No Coverage | | | | | | | | | | |

Displaying (1 of 1)

**Last test execution:** Pass

# 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

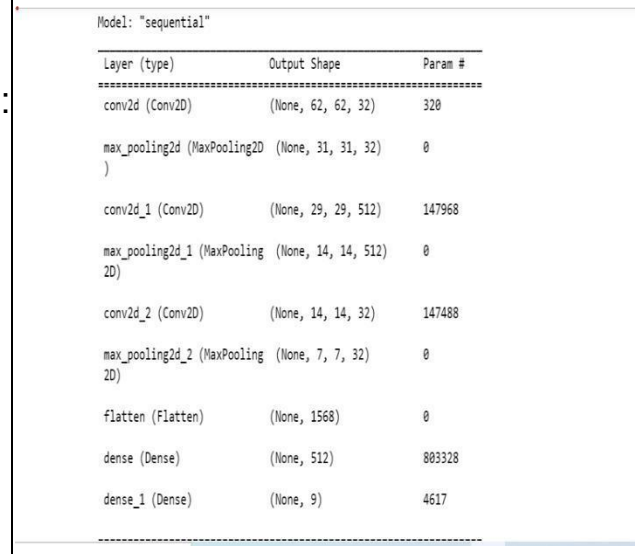
| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Subtotal |
|---|---|---|---|---|---|
| By Design | 11 | 7 | 4 | 2 | 24 |
| Duplicate | 1 | 0 | 2 | 0 | 3 |
| External | 2 | 3 | 2 | 1 | 8 |
| Fixed | 10 | 5 | 3 | 14 | 32 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 1 | 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

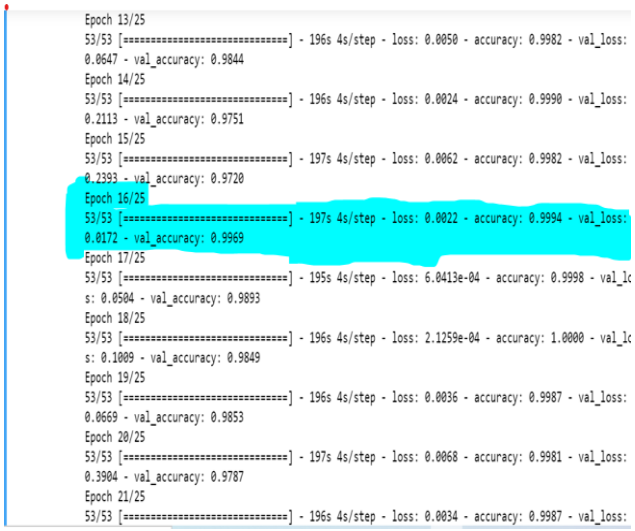
| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| Print Engine | 7 | 0 | 0 | 7 |
| Client Application | 15 | 0 | 0 | 15 |
| Security | 2 | 0 | 0 | 2 |
| Outsource Shipping | 2 | 0 | 0 | 2 |
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version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

## .3 Performance Testing

| S.NO | Parameter | Values | Screenshot |
|---|---|---|---|
| 1. | Model Summary | Total params: 1,103,721<br>Trainable params: 1,103,721<br>Non-trainable params: 0 |  |

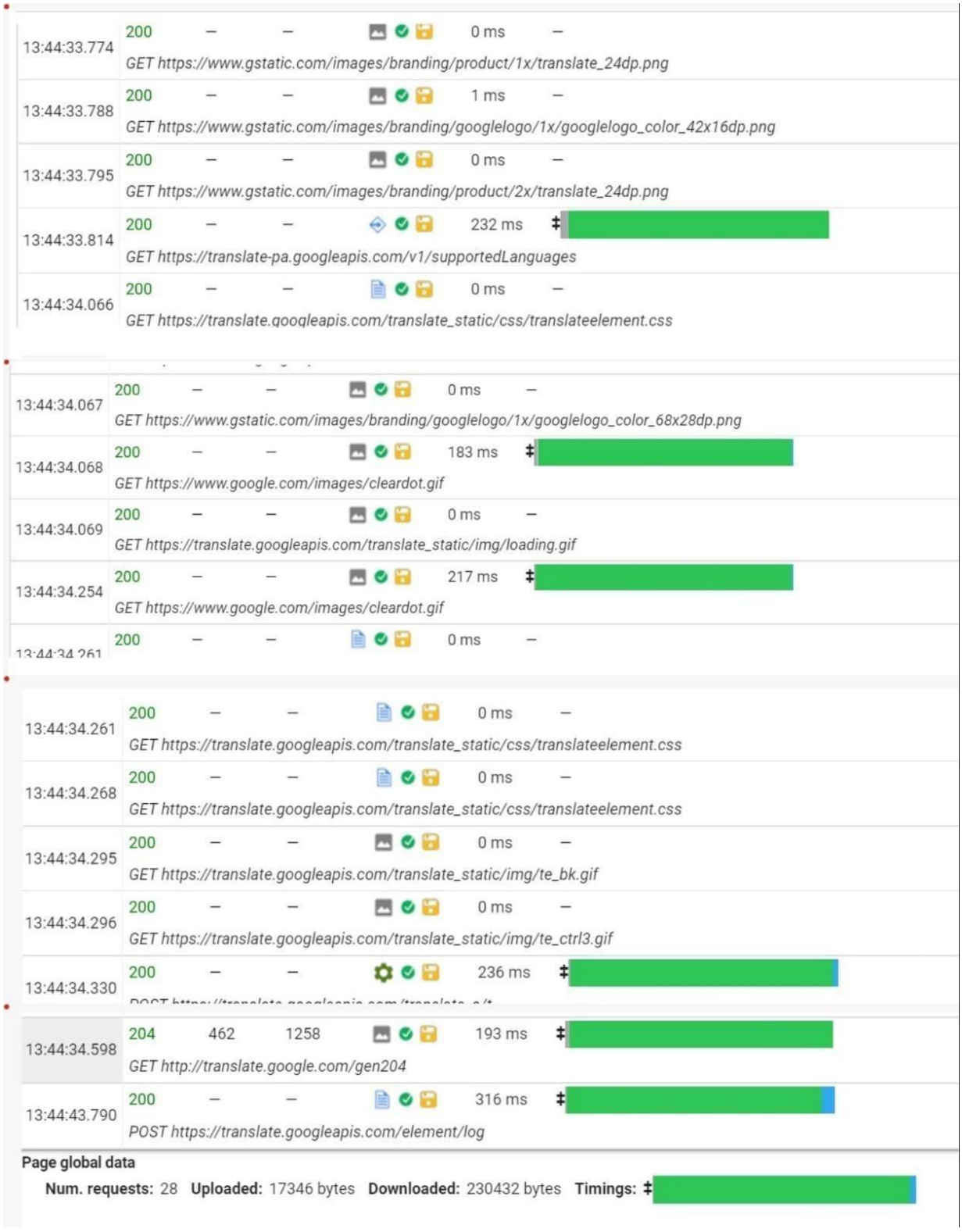
Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 62, 62, 32) | 320 |
| max_pooling2d (MaxPooling2D) | (None, 31, 31, 32) | 0 |
| conv2d_1 (Conv2D) | (None, 29, 29, 512) | 147968 |
| max_pooling2d_1 (MaxPooling2D) | (None, 14, 14, 512) | 0 |
| conv2d_2 (Conv2D) | (None, 14, 14, 32) | 147488 |
| max_pooling2d_2 (MaxPooling2D) | (None, 7, 7, 32) | 0 |
| flatten (Flatten) | (None, 1568) | 0 |
| dense (Dense) | (None, 512) | 803328 |
| dense_1 (Dense) | (None, 9) | 4617 |

| | | | |
|---|---|---|---|
| 2. | Accuracy | Training Accuracy - 0.9994<br><br>Validation Accuracy -0.9969 |  |

# 9. Results  Performance Metrics

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


13:44:33.774 200 – – 0 ms –
GET https://www.gstatic.com/images/branding/product/1x/translate_24dp.png
13:44:33.788 200 – – 1 ms –
GET https://www.gstatic.com/images/branding/googlelogo/1x/googlelogo_color_42x16dp.png
13:44:33.795 200 – – 0 ms –
GET https://www.gstatic.com/images/branding/product/2x/translate_24dp.png
13:44:33.814 200 – – 232 ms
GET https://translate-pa.googleapis.com/v1/supportedLanguages
13:44:34.066 200 – – 0 ms –
GET https://translate.googleapis.com/translate_static/css/translateelement.css
13:44:34.067 200 – – 0 ms –
GET https://www.gstatic.com/images/branding/googlelogo/1x/googlelogo_color_68x28dp.png
13:44:34.068 200 – – 183 ms
GET https://www.google.com/images/cleardot.gif
13:44:34.069 200 – – 0 ms –
GET https://translate.googleapis.com/translate_static/img/loading.gif
13:44:34.254 200 – – 217 ms
GET https://www.google.com/images/cleardot.gif
200 – – 0 ms –
13:44:34.261 200 – – 0 ms –
GET https://translate.googleapis.com/translate_static/css/translateelement.css
13:44:34.268 200 – – 0 ms –
GET https://translate.googleapis.com/translate_static/css/translateelement.css
13:44:34.295 200 – – 0 ms –
GET https://translate.googleapis.com/translate_static/img/te_bk.gif
13:44:34.296 200 – – 0 ms –
GET https://translate.googleapis.com/translate_static/img/te_ctrl3.gif
13:44:34.330 200 – – 236 ms
13:44:34.598 204 462 1258 193 ms
GET http://translate.google.com/gen204
13:44:43.790 200 – – 316 ms
POST https://translate.googleapis.com/element/log
Page global data
Num. requests: 28 Uploaded: 17346 bytes Downloaded: 230432 bytes Timings:

# 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 Model Building

```
import cv2 import os os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2' import numpy as np from keras.models import Sequential import matplotlib.pyplot as plt from keras.layers import Dense, Dropout, Activation, Flatten from keras.layers import Conv2D, MaxPool2D from keras_preprocessing.image import ImageDataGenerator test_path = 'Dataset/test_set' train_path
= 'Dataset/training_set'
train=ImageDataGenerator(rescale=1./255,zoom_range=0.2,shear_range=0.2,horizontal_flip=T rue)
test=ImageDataGenerator(rescale=1./255) train_batches =
train.flow_from_directory(directory=train_path, target_size=(64,64), class_mode='categorical', batch_size=300,shuffle=True,color_mode="grayscale") test_batches =
test.flow_from_directory(directory=test_path, target_size=(64,64), class_mode='categorical', batch_size=300, shuffle=True,color_mode="grayscale") model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(64,64,1)))
model.add(MaxPool2D(pool_size=(2,2))) model.add(Conv2D(512, (3, 3),
```

padding="valid")) model.add(MaxPool2D(pool_size=(2,2))) model.add(Conv2D(32,

(3, 3), padding="same")) model.add(MaxPool2D(pool_size=(2,2))) model.add(Flatten())
model.add(Dense(512,activation ="relu")) model.add(Dense(9,activation ="softmax"))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy']) history

= model.fit(train_batches, batch_size=32,validation_data=test_batches,epochs=25)

model.save('model.h5')

# Model Testing

import keras from keras.models import
load_model import cv2 import numpy as np import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
val=['A','B','C','D','E','F','G','H','I']

model=load_model('model.h5') from skimage.transform import resize def detect(frame):
img=resize(frame,(64,64,1)) img=np.expand_dims(img,axis=0)
if(np.max(img)>1): img = img/255.0 predict_x=model.predict(img) print(predict_x)
predict=np.argmax(predict_x,axis=1) x=predict[0] print(val[x])
frame=cv2.imread(r"C:\Users\Akshaya\PycharmProjects\Realtime_Communicati
on_System_For_Specially_Abled\Dataset\test_set\B\1.png") data=detect(frame)

# Flask App Building

from flask import Flask, Response, render_template
import cv2

```
app =
Flask(__name__)
cap =
cv2.VideoCapture(
0)
@app.rou
te('/') def
index():
   return  render_template('index.html')

def
generate_frames():
while True:
      success, frame = cap.read()
imgOutput=frame.copy()        yield (b'--frame\r\n'
b'Content-Type: image/jpeg\r\n\r\n' + imgOutput + b'\r\n')
@app.route('/predict',methods=['POST','G
ET']) def predictions():
   #The prediction model code goes here
   #Once the start Button is pressed the prediction model
starts    pass
@app.route('/stop',methods=['POST','G
ET']) def stopping():
   #The text to speech code goes here
   #Once the stop button is pressed the text is converted into
speech    pass
```

```
@app.route('/video')
def video():
   return Response(generate_frames(),mimetype='multipart/x-mixed-replace;
boundary=frame')



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

# HTML Files

**index.html** <!DOCTYPE

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


   <h2 class="header">Sign Language TO Speech</h2>
   <div class="video">
      <img src="{{ url_for('video') }}" width="50%">
   </div>
   <div class="container">
   <form action='/predict' method='post'>
   <button type="submit" name="start" value="start" class="button1"
>Start</button>     </form>
 <form action='/stop' method='post'>

   <button type="submit" name="stop" value="stop" class="button2"
>Stop</button>   </form>
 </div>
 <div class="instruction">
   <center>
   <p> <details>
     <summary><b>Instructions to Use</b></summary>
```

```
    <p>Once the webcam is <b>ON</b> Click <strong>"START"</strong> to start
the predicition model.<br>
    <br>
    >>Click <strong>"s"</strong> to save the
text.<br>      <br>
    >>Click <strong>"a"</strong> to leave a space.<br>
    <br>
    >>Click <strong>"d"</strong> to delete a character from right to
left.<br>      <br>
    >>Click <strong>"w"</strong> to delete entire
text.<br>      <br>
    >> The Saved text appears on the top left corner of the video
Screen<br>      <br>
    >> Once you are satisfied with the saved text press<b>"STOP"</b> to convert it
into speech<br>
    <br>
    >><b>NOTE: The hand must be on the screen to display the text to save,delete
or to leave a space between them.</b>

    </p>
   </details></p>
  </center>
</div>
<br>
<br>
<div class="team">
<center>
<details>
  <summary><b>Team</b></summary>
  <p><b>Ibm ID-2475-1658472446</b>
  <br>
   1.Sajith<br>
   2.Stanlee<br>
   3.Sachin<br>
   4.Harish<br>
```

```
  </p>
</details>
</center>
</div>
<p>Feel Free to contact us !!!!!</p>
<center>
<div class="alert info">
  <span class="closebtn">&times;</span>
  <strong>NOTE:</strong> A disturbance free background with good
lighting(Whitebackground) is preferred.

</div>
</center>
<script>
  var close =
document.getElementsByClassName("closebtn");   var i;

  for (i = 0; i <
close.length; i++) {
close[i].onclick =
function(){       var div =
this.parentElement;
div.style.opacity = "0";
     setTimeout(function(){ div.style.display = "none"; }, 600);
    }
  }
  </script>

  <a class="social-icon"href="https://in.linkedin.com/in/sajith-m-
82431721a?trk=profilebadge" target="_blank">
    <ion-icon name="logo-linkedin"></ion-icon>
  </a>
  <a class="git"href="https://github.com/IBM-EPBL/IBM-Project-2475-
1658472446.git" target="_blank">
    <ion-icon name="logo-github"></ion-icon>
  </a>
```

```
  <script src="https://platform.linkedin.com/badges/js/profile.js"
async defer type="text/javascript"></script>   <script
type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
```

<script nomodule

```
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>



  </body>
</html>
```

**Speech_to_sign,html**

```
<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<script
src="http://translate.google.com/translate_a/element.js?cb=loadGoogleTranslate"
></script> <script> function loadGoogleTranslate()

{

new google.translate.TranslateElement("google_element")

}

</script>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></scri
pt>
```

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<style> .row { display: flex; }

.col { flex:
50%;
}
*,*:after,*:before{
-webkit-box-sizing: border-box;  -moz-box-sizing: border-box; -ms-boxsizing: border-box; box-sizing: borderbox;

} body{  font-family: arial; font-size: 16px; margin: 0; color: #000; display: flex; align-items: center; justify-content: center; min-height: 100vh; }

.voice_to_text{ width: 600px; textalign: center; } h1{ color: #000000; fontsize:

```
50px; } #convert_text{ width: 100%; height: 200px; border-radius: 10px; resize: none; padding: 10px; fontsize: 20px; marginbottom: 10px; } button{  padding: 12px 20px; background: #0ea4da; border: 0; color: #fff; font-size: 18px; cursor: pointer; border-radius: 5px; }

</style>  </head>

<body>
<div class="container">
<div class="row">
<div class="col">
<img src="https://img.freepik.com/free-vector/sign-language-alphabethanddrawn-style_23-2147872270.jpg?w=2000" style="width:50%;"/>
</div>
<div class="col"><div class="voice_to_text">
<div class="text_center" id="google_element"></div>
<h1>Voice to text converter</h1>
<textarea name="" id="convert_text"></textarea>
<button id="click_to_record" class="btn-primary">Voice to Text</button><br/>
<a href="/">
<button class="btn btn-danger btn-lg">Exit</button>
</a>
</div>
```

<script type="text/javascript" src="{{ url_for('static', filename='javascript/script.js') }}"></script> </body>

</html>

Sign_to_speech.html

<html>  <head> <style> img{ display: block; margin-left: auto; margin-right: auto;

}

</style> <script> function loadGoogleTranslate()

{

new google.translate.TranslateElement("google_element")

}

</script>

<script src="http://translate.google.com/translate_a/element.js?cb=loadGoogleTranslate" ></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"/>
</head>

<body>

```
<h1>Sign to speech</h1>
<div>
<div class="text-center" id="google_element"></div>
<img src="{{ url_for('video') }}" width="50%" height="50%"/>
<br/>
<div class="text-center">
<a href="/">
<button class="btn btn-danger btn-lg" >Exit</button>
</a></div>
</div>

</body>

</html>
```

# CSS Files

**Index.css**

```
@import
url("https://fonts.googleapis.com/css2?family=Oxygen:wght@400;700&family=Ro
boto:w ght@300;900&display=swap");

* {
  box-sizing:
border-box;
padding:        0;
margin: 0;
}

:root {
  --black: #000;
  --white: #fff;
  --hover: #000;
```

```
}

.main {
position:
relative;
height:
100vh;
  width: 100%;

  display: flex;
align-items:
center;   justify-
content: center;
}

.inside {
position:
relative;
height:
60%;
width:
50%;
  background: rgba(255,255,255,0.9);
border-radius: 30px;
  /* border: 5px solid var(--
black); */   display: flex;
align-items: center;
  justify-content: space-evenly;
  -webkit-box-shadow: 12px 12px 17px 1px rgba(0, 0, 0, 0.59);
  -moz-box-shadow: 12px 12px 17px 1px rgba(0, 0, 0, 0.59);
box-shadow: 12px 12px 17px 1px rgba(0, 0, 0, 0.59);
}

.wrapper {   position:
relative;   height:
75%;   width: 30%;
display: flex;   align-
items: center;
```

```
justify-content:
space-evenly;   flex-
direction: column;
}

.Head {   position:
relative;   font-size: 3rem;
text-transform:
uppercase;   font-family:
"Roboto", sans-serif;
font-weight: 900;
display: flex;   align-
items: center;   justify-
content: center;   flex-
direction: column;
height: 30%;
}

.Head h1 {
font-size:
3rem;
}

.Head span {
position: relative;
height: 5px;
width: 60%;
background: var(-
-black); }

.box {   position: relative;
font-family: "Oxygen",
sans-serif;   font-weight:
700;   border: 2px solid
var(--black);   border-
radius: 1.5rem;   text-
decoration: none;
```

```
overflow: hidden;   cursor:
pointer;   z-index: 1;
}

.box1 {
  padding: 0.8rem 2rem;
}

.box2 {
  padding: 0.8rem 1.5rem;
}

.box:hover {
color: var(--
white);
background: var(-
-hover);
}
```

# Javascript Files

**Script.js**

```
click_to_record.addEventListener('click',function(
){     var speech = true;     const
SpeechRecognition = window.speechRecognition
|| window.webkitSpeechRecognition;


   const recognition =new SpeechRecognition();
recognition.interimResults = true;
recognition.addEventListener('result', e => {
```

```
        const transcript = Array.from(e.results)
            .map(result => result[0])
            .map(result => result.transcript)
            .join('')

        document.getElementById("convert_text").innerHTML = transcript;
console.log(transcript);
    });


    if (speech == true) {
recognition.start();
    }
})
```

## Output

## 13.2 Github and Demo Link:

https://github.com/IBM-EPBL/IBM-Project-28638-1660114698