Real-Time Communication System Powered by AI for Specially Abled

TEAM ID: PNT2022TMID05416

Submitted by

BALAMURUGAN P 921319205014

ARUNACHALAA A.S 921319205008

ARUNPANDIAN J 921319205011

KAILASH A.D 921319205052

In partial fulfilment for the

award of the degree   of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY

DINDIGUL- 624622

Table Of Contents

1.INTRODUCTION

1.1 Project Overview

Speech impaired people use hand signs and gestures to communicate. Normal people face difficulty in understanding their language. Hence there is a need of a system which recognizes the different signs, gestures and conveys the information to the normal people. It bridges the gap between physically challenged people and normal people

1.2 Purpose

Sign language is the only tool of communication for the person who is not able to speak and hear anything. Sign language is a boon for the physically challenged people to express their thoughts and emotion. In this work, a novel scheme of sign language recognition has been proposed for identifying the alphabets and gestures in sign language. With the help of computer vision and neural networks we can detect the signs and give the respective text output.

2. LITERATURESURVEY

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 Reference:

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.

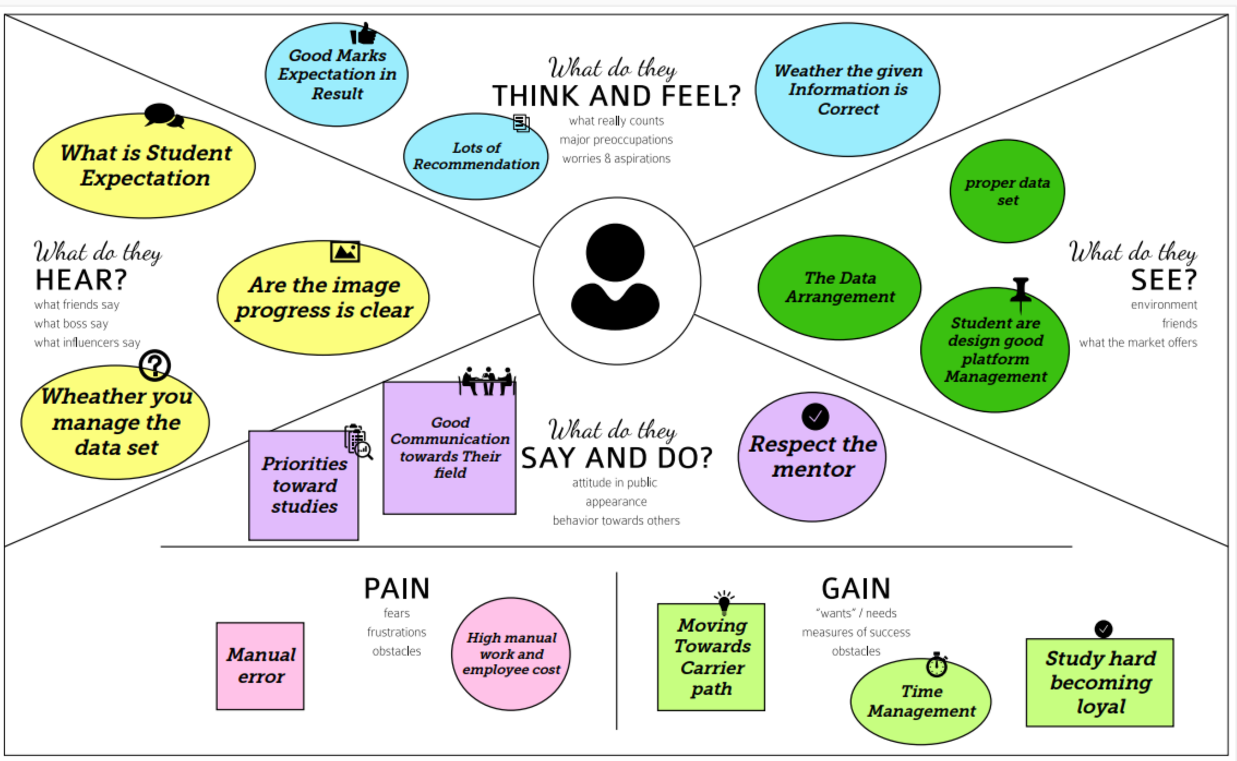
2.3 Problem Statement Definition:

Speech impaired people use hand signs and gestures to communicate.

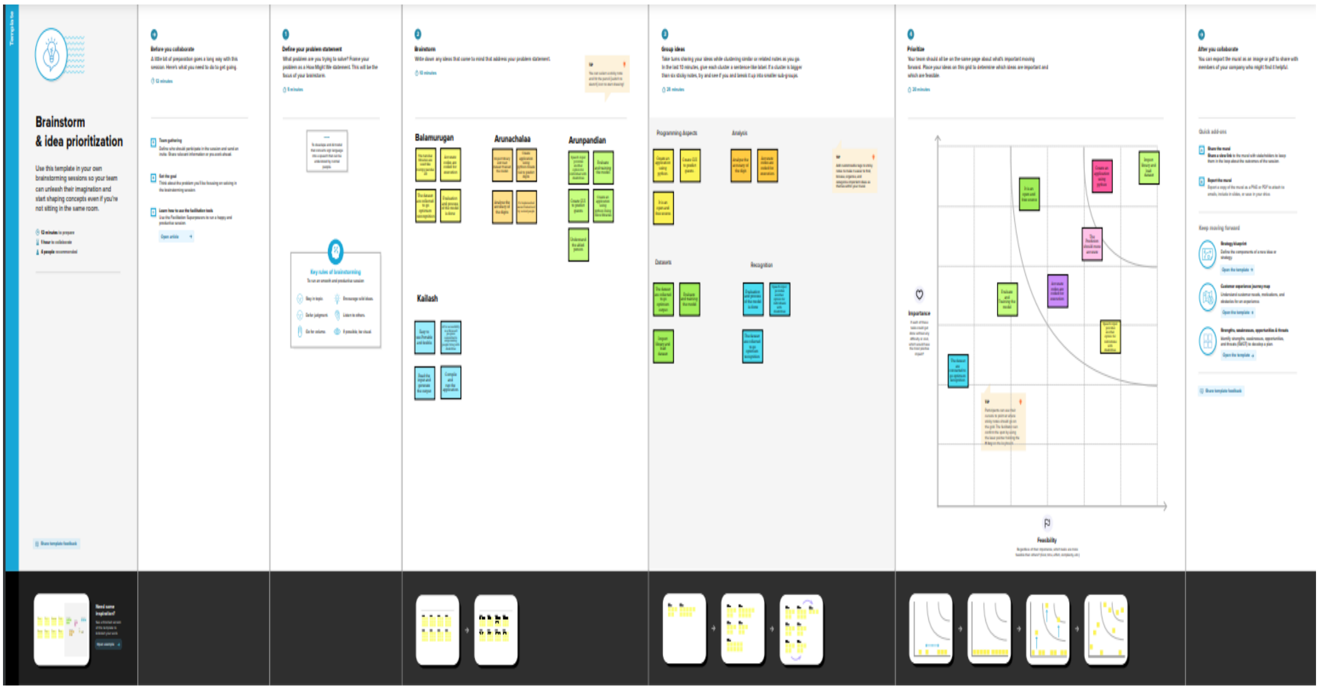
Normal people face difficulty in understanding their language. Hence there is a need of a system which recognizes the different signs, gestures and conveys the information to the normal people. It bridges the gap between physically challenged people and normal people.

3. Ideation and Proposed Solution

3.1 Empathy Map Canvas:



3.2 Ideation and Brainstorming:


Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Before you collaborate

Define your problem statement

Key rules of brainstorming

Brainstorm

Balamurugan

Arunachalaa

Arunpandian

Kailash

Group ideas

Programming Aspects

Datasets

Prioritize

Importance

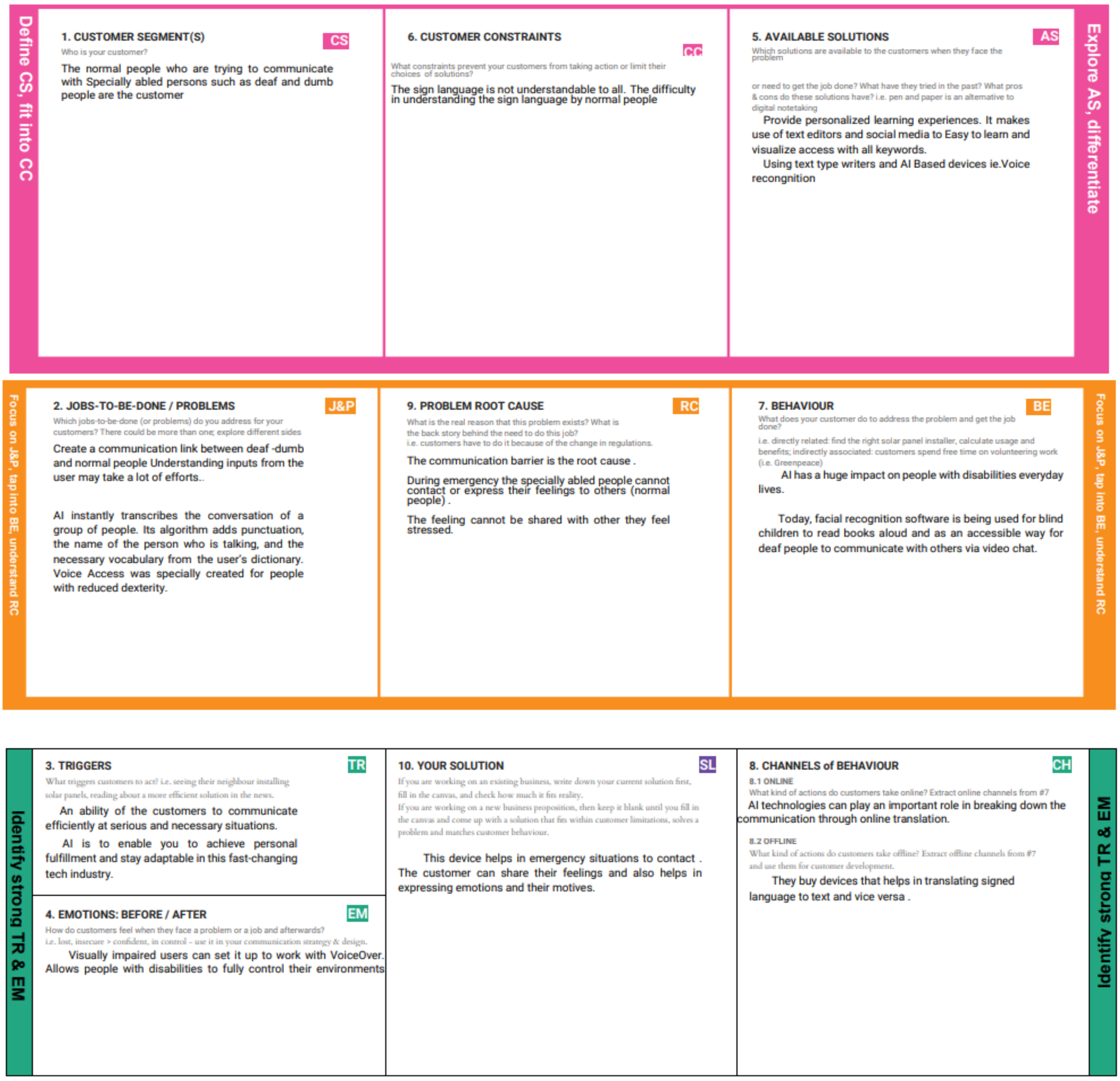
Feasibility

After you collaborate

3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Everyone is not convenient with language used in the application . • Some people cannot understand English we can convert into their convenient language . • They are facing difficulties in understanding the language used in the system.
2.	Idea / Solution description	• Even sign language can also be translated to text message in our application using CNN. • Text to sign language convertor uses Stanford Parser text processing and JA Signing for the signing avatar. • Can change the language using google language translator tool so that people can use the application based on their specialized language. • Producing a model which can recognize Finger-spelling based hand gestures in order to form a complete word by combining each gesture. • By using this application both specially abled and normal people can translate their messages to others easily.
3.	Novelty / Uniqueness	• This model using SSD ML algorithm recognizing the signs as words instead of old traditional translators, that are very slow and take too much since every alphabet as to be recognized to form the whole statement in old methods. • Normal text can also be translated into sign language.
4.	Social Impact / Customer Satisfaction	• The deaf and dump people can easily translate their sign language into a human hearing voice. • The normal people can also easily translate their voice into a sign language using this application
5.	Business Model (Revenue Model)	• We can generate revenue by offering subscription- For unlimited usage and Ad free. • Users who have got subscription can change the language accordingly
6.	Scalability of the Solution	• The model which is TensorFlow model that has been used can be replaced with another model as well. • The same system can be implemented for different sign languages by substituting the dataset.

3.4 Proposed Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

The normal people who are trying to communicate with Specially abled persons such as deaf and dumb people are the customer

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

The sign language is not understandable to all. The difficulty in understanding the sign language by normal people

5. AVAILABLE SOLUTIONS AS

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

or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Provide personalized learning experiences. It makes use of text editors and social media to Easy to learn and visualize access with all keywords.

Using text type writers and AI Based devices ie.Voice recongnition

Explore AS, differentiate

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

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

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

Create a communication link between deaf -dumb and normal people Understanding inputs from the user may take a lot of efforts..

AI instantly transcribes the conversation of a group of people. Its algorithm adds punctuation, the name of the person who is talking, and the necessary vocabulary from the user's dictionary. Voice Access was specially created for people with reduced dexterity.

9. PROBLEM ROOT CAUSE RC

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

The communication barrier is the root cause .

During emergency the specially abled people cannot contact or express their feelings to others (normal people) .

The feeling cannot be shared with other they feel stressed.

7. BEHAVIOUR BE

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

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

AI has a huge impact on people with disabilities everyday lives.

Today, facial recognition software is being used for blind children to read books aloud and as an accessible way for deaf people to communicate with others via video chat.

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

Identify strong TR & EM

3. TRIGGERS TR

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

An ability of the customers to communicate efficiently at serious and necessary situations.

AI is to enable you to achieve personal fulfillment and stay adaptable in this fast-changing tech industry.

4. EMOTIONS: BEFORE / AFTER EM

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

Visually impaired users can set it up to work with VoiceOver. Allows people with disabilities to fully control their environments

10. YOUR SOLUTION SL

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

This device helps in emergency situations to contact . The customer can share their feelings and also helps in expressing emotions and their motives.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

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

AI technologies can play an important role in breaking down the communication through online translation.

8.2 OFFLINE

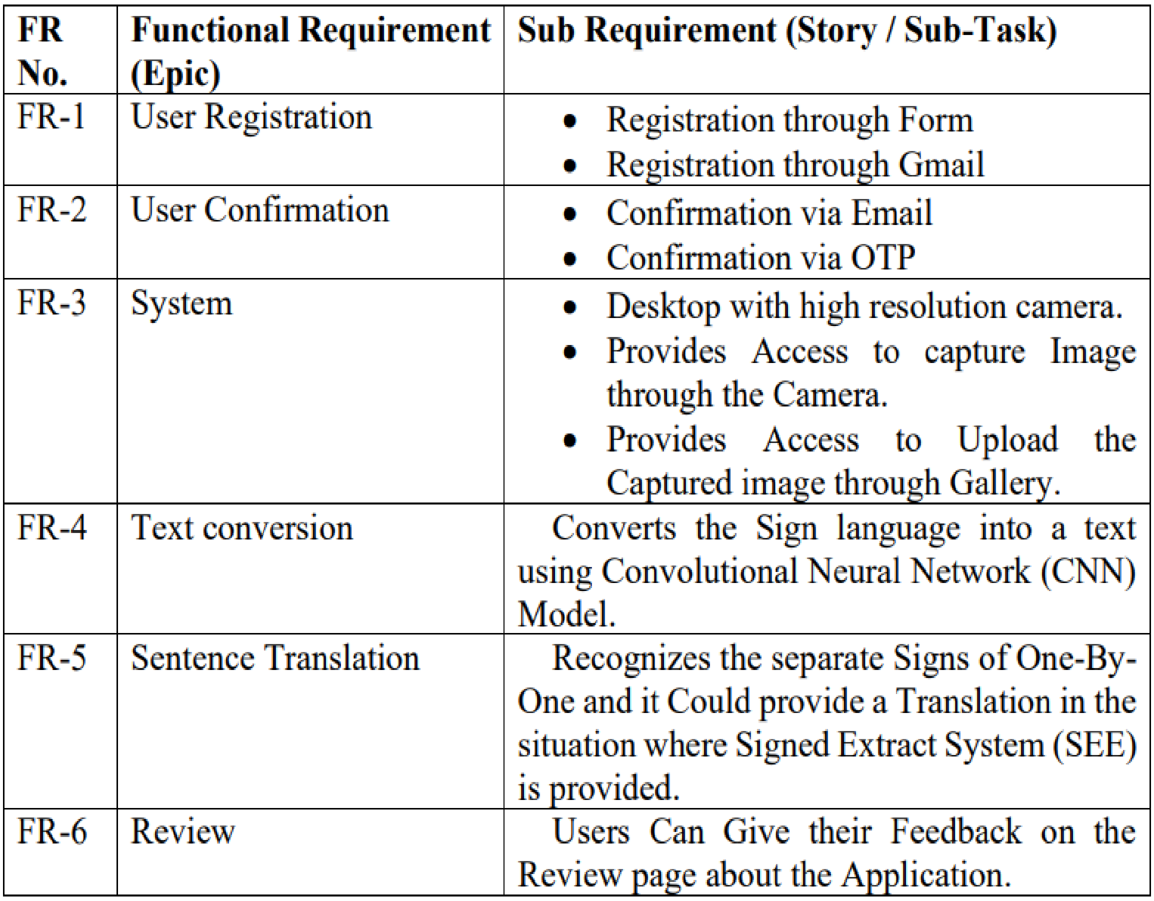
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

They buy devices that helps in translating signed language to text and vice versa .

Identify strong TR & EM

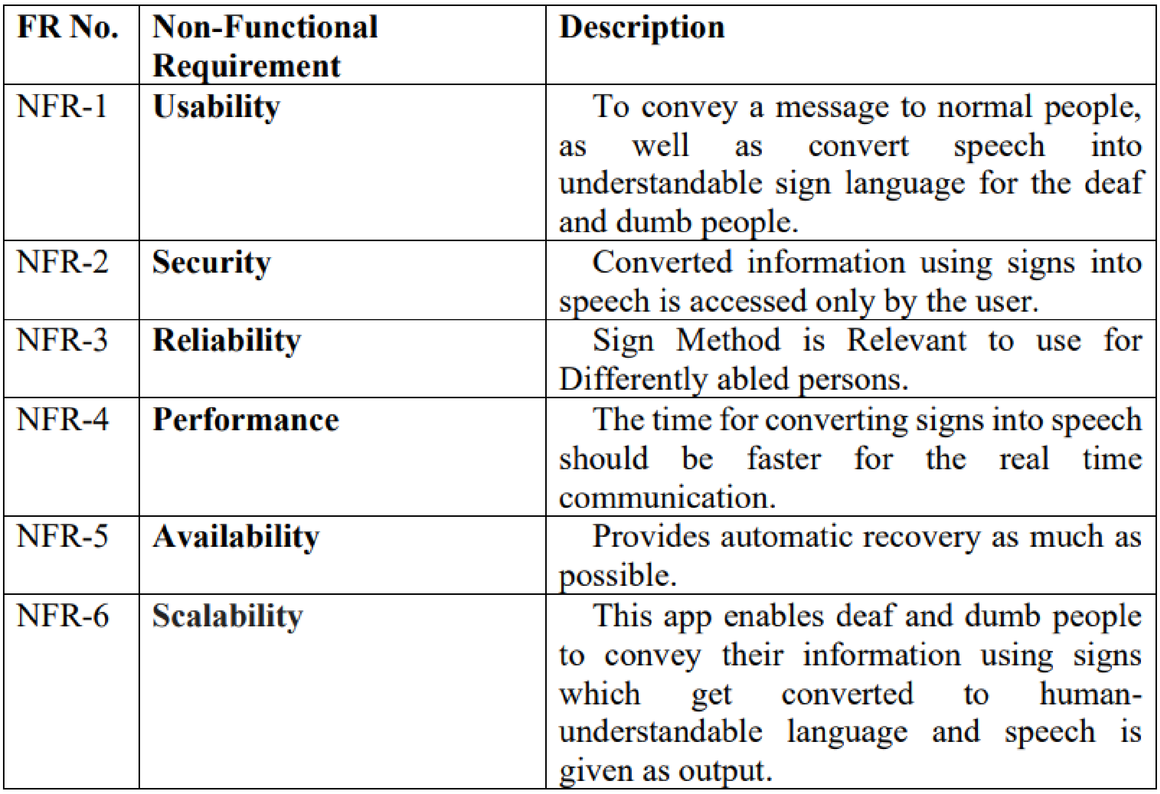
<u>4. Requirement Analysis</u>

4.1 Functional Requirement:



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	• Registration through Form • Registration through Gmail
FR-2	User Confirmation	• Confirmation via Email • Confirmation via OTP
FR-3	System	• Desktop with high resolution camera. • Provides Access to capture Image through the Camera. • Provides Access to Upload the Captured image through Gallery.
FR-4	Text conversion	Converts the Sign language into a text using Convolutional Neural Network (CNN) Model.
FR-5	Sentence Translation	Recognizes the separate Signs of One-By-One and it Could provide a Translation in the situation where Signed Extract System (SEE) is provided.
FR-6	Review	Users Can Give their Feedback on the Review page about the Application.

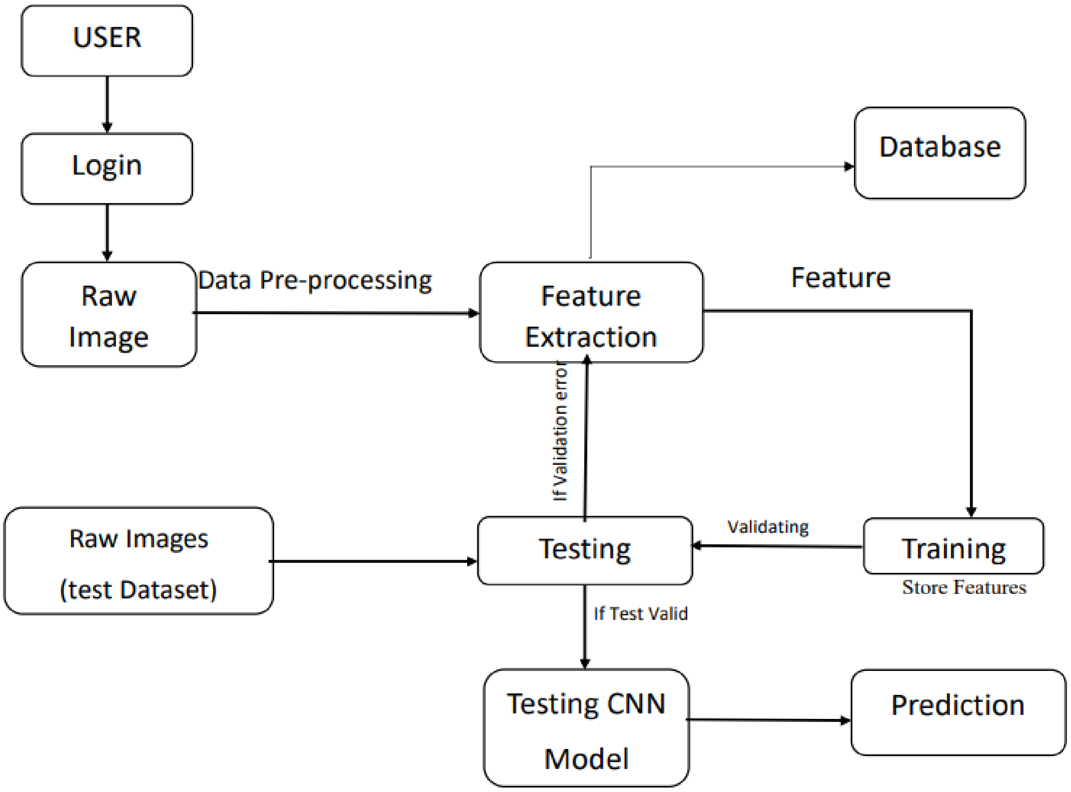
4.2 Non-Functional Requirement:



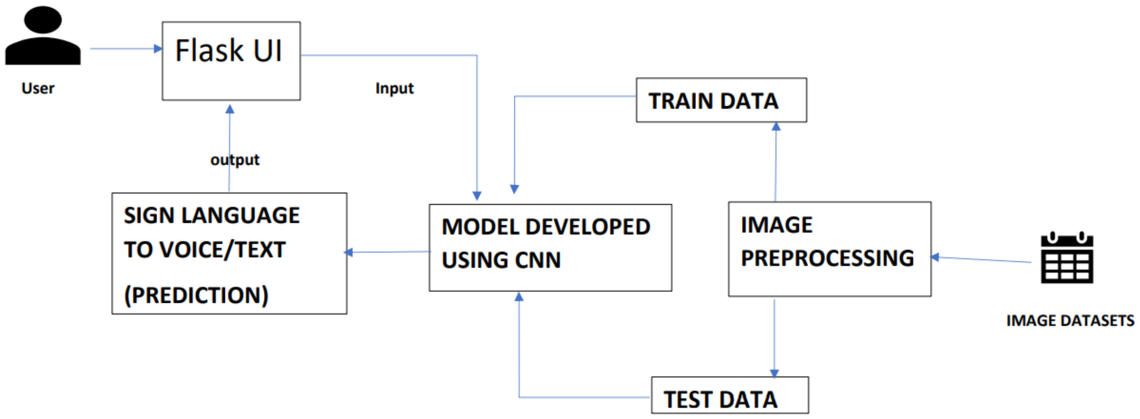
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	To convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb people.
NFR-2	**Security**	Converted information using signs into speech is accessed only by the user.
NFR-3	**Reliability**	Sign Method is Relevant to use for Differently abled persons.
NFR-4	**Performance**	The time for converting signs into speech should be faster for the real time communication.
NFR-5	**Availability**	Provides automatic recovery as much as possible.
NFR-6	**Scalability**	This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

5. Project Design

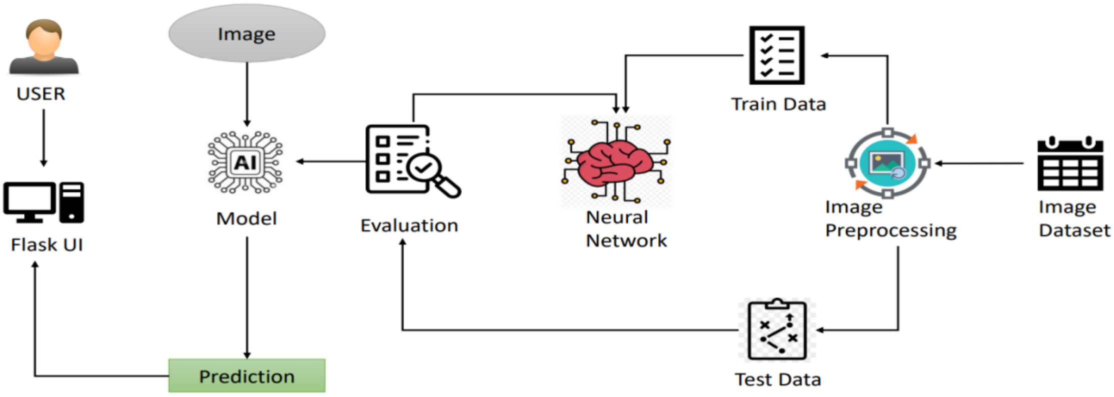
5.1 DATA FLOW DIAGRAM:



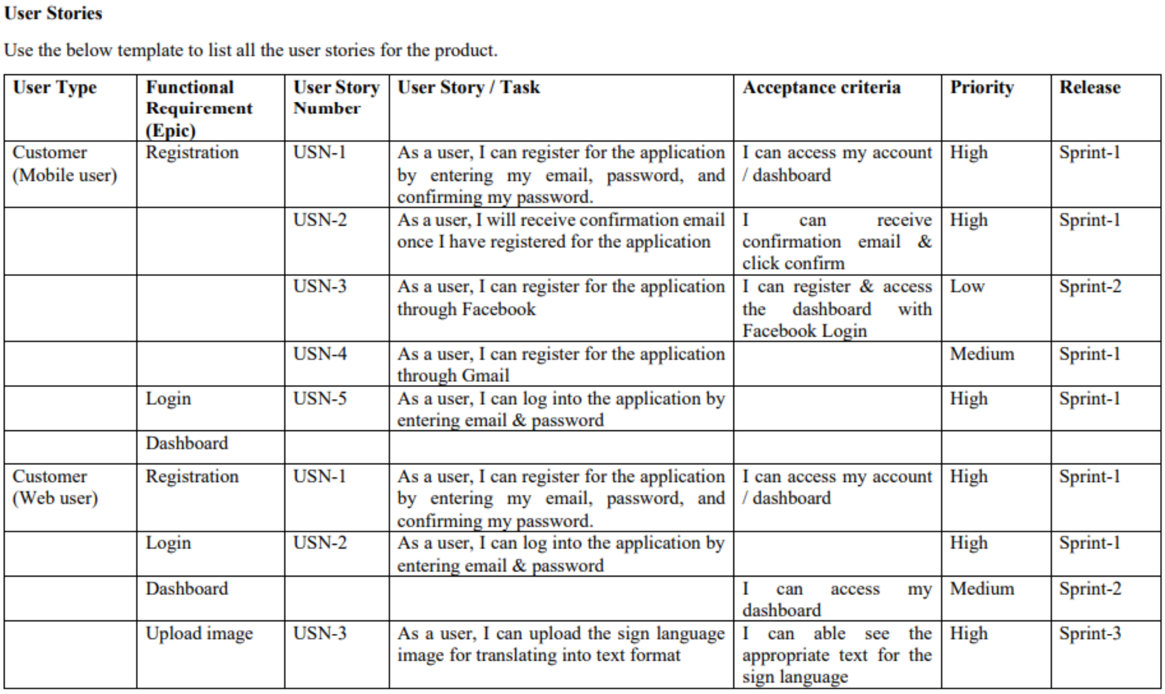
5.2.1 Solution Architecture:



5.2.2 Technical Architecture:



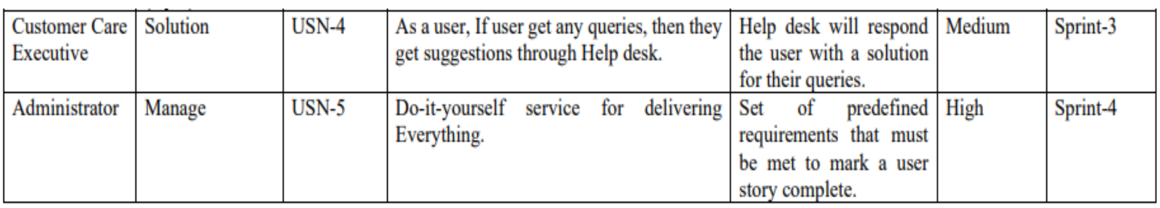
5.3 User Stories:



User Stories

Use the below template to list all the user stories for the product.

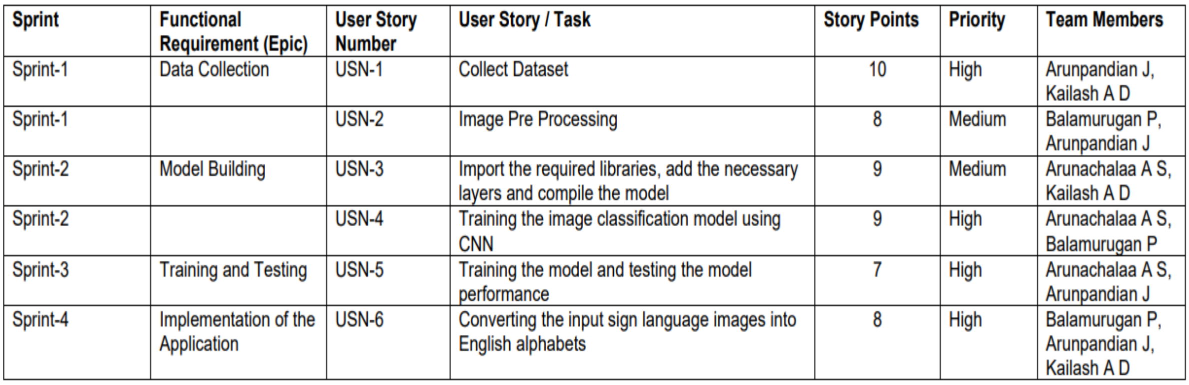
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)	Registration	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	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard			I can access my dashboard	Medium	Sprint-2
	Upload image	USN-3	As a user, I can upload the sign language image for translating into text format	I can able see the appropriate text for the sign language	High	Sprint-3



Customer Care Executive	Solution	USN-4	As a user, If user get any queries, then they get suggestions through Help desk.	Help desk will respond the user with a solution for their queries.	Medium	Sprint-3
Administrator	Manage	USN-5	Do-it-yourself service for delivering Everything.	Set of predefined requirements that must be met to mark a user story complete.	High	Sprint-4

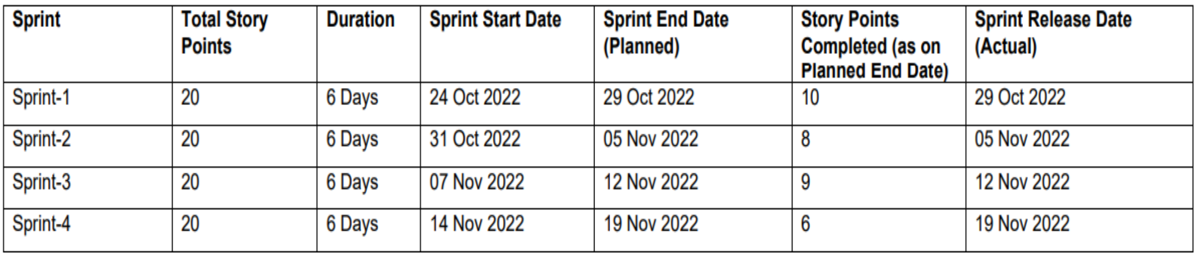
6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:



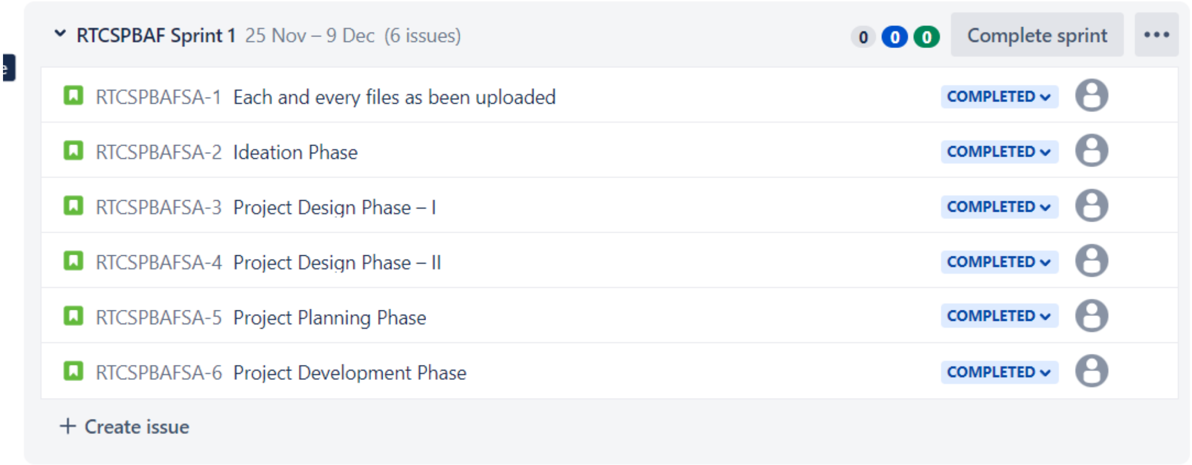
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect Dataset	10	High	Arunpandian J, Kailash A D
Sprint-1		USN-2	Image Pre Processing	8	Medium	Balamurugan P, Arunpandian J
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	9	Medium	Arunachalaa A S, Kailash A D
Sprint-2		USN-4	Training the image classification model using CNN	9	High	Arunachalaa A S, Balamurugan P
Sprint-3	Training and Testing	USN-5	Training the model and testing the model performance	7	High	Arunachalaa A S, Arunpandian J
Sprint-4	Implementation of the Application	USN-6	Converting the input sign language images into English alphabets	8	High	Balamurugan P, Arunpandian J, Kailash A D

6.2 Sprint Delivery Schedule:



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	8	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	9	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	6	19 Nov 2022

6.3 Report From JIRA:



RTCSPBAF Sprint 1 25 Nov – 9 Dec (6 issues)

RTCSPBAFSA-1	Each and every files as been uploaded	COMPLETED
RTCSPBAFSA-2	Ideation Phase	COMPLETED
RTCSPBAFSA-3	Project Design Phase – I	COMPLETED
RTCSPBAFSA-4	Project Design Phase – II	COMPLETED
RTCSPBAFSA-5	Project Planning Phase	COMPLETED
RTCSPBAFSA-6	Project Development Phase	COMPLETED

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 Feature 1(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 Feature 2(Real time speech to text):

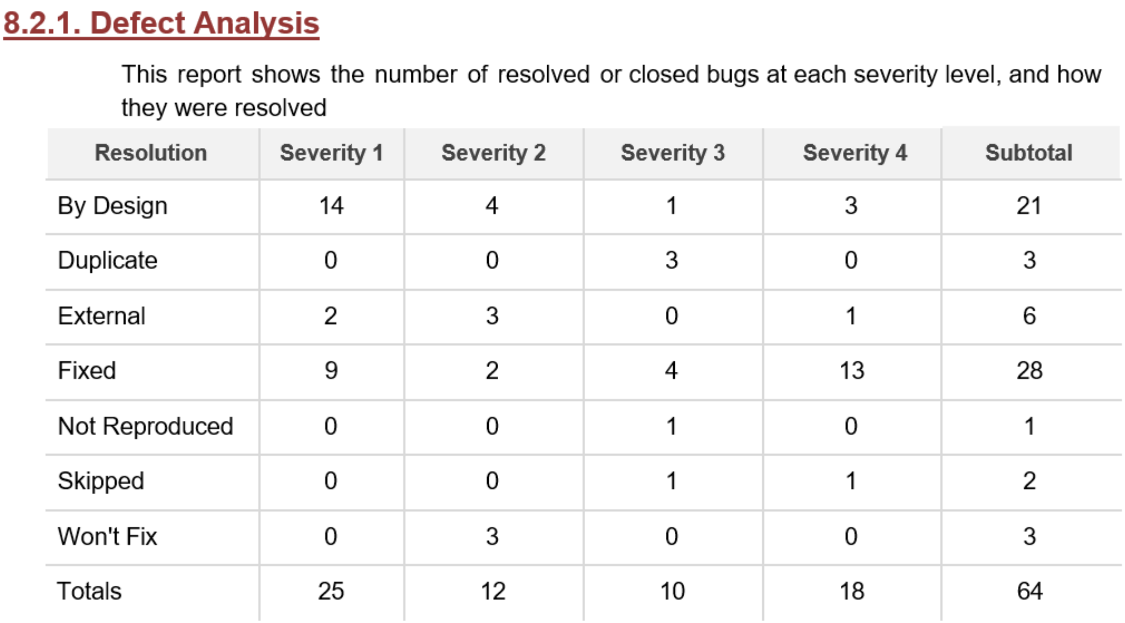
With the Web Speech API, we can recognize speech using JavaScript. It is super easy to recognize speech in a browser using JavaScript and then get the text from the speech to use as user input. We use the Speech Recognition object to convert the speech into text and then display the text on the screen. 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 Testcases:

- 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 User Acceptance Testing(UAT):



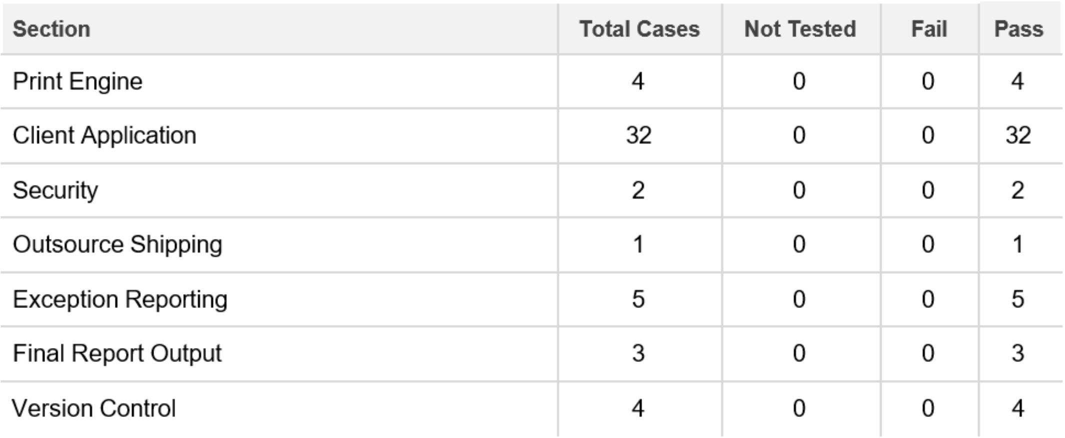
8.2.1. Defect Analysis

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	14	4	1	3	21
Duplicate	0	0	3	0	3
External	2	3	0	1	6
Fixed	9	2	4	13	28
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	3	0	0	3
Totals	25	12	10	18	64

8.2.2. Test Case Analysis

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

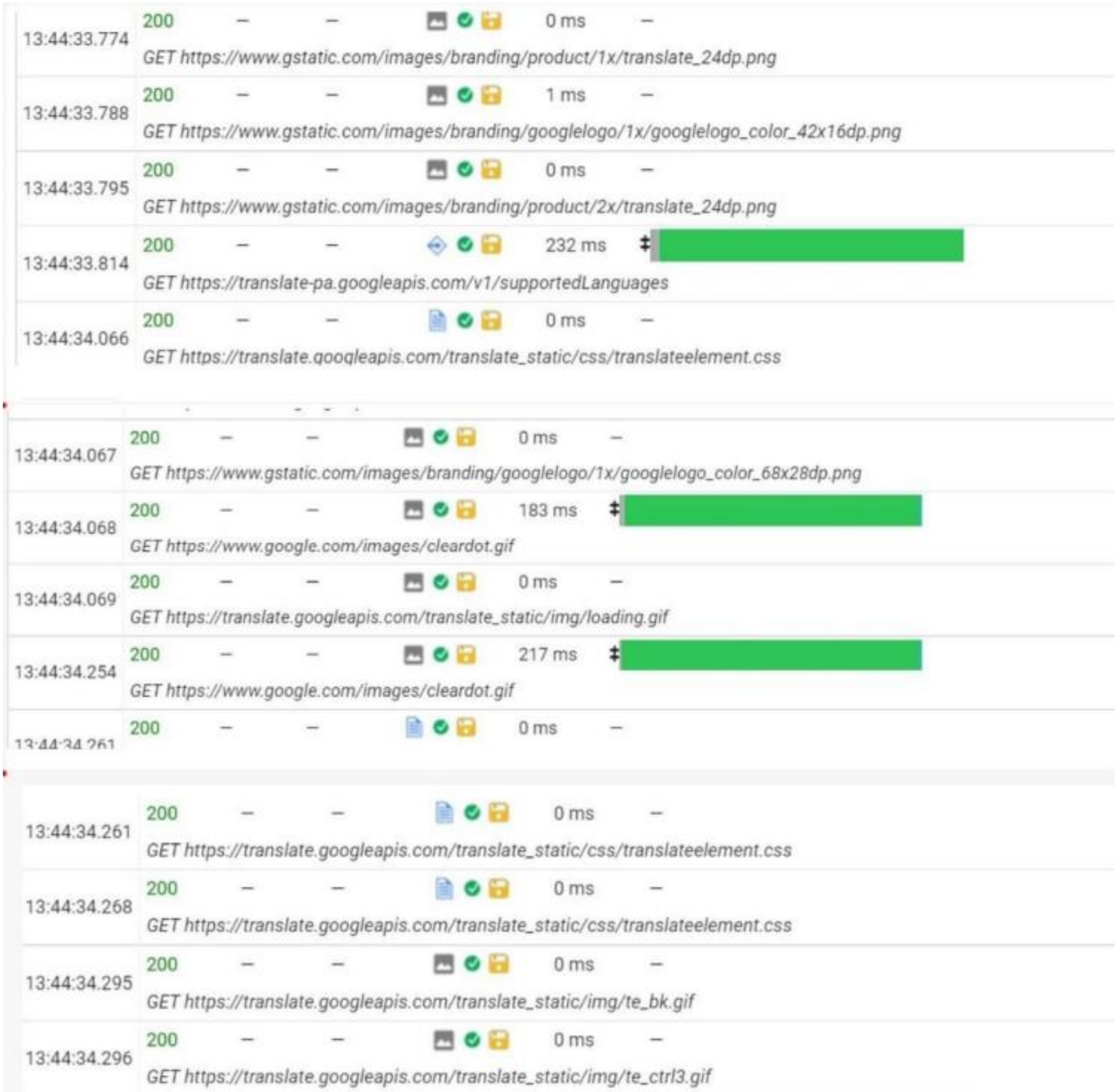


Section	Total Cases	Not Tested	Fail	Pass
Print Engine	4	0	0	4
Client Application	32	0	0	32
Security	2	0	0	2
Outsource Shipping	1	0	0	1
Exception Reporting	5	0	0	5
Final Report Output	3	0	0	3
Version Control	4	0	0	4

9.RESULTS

9.1 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
13:44:34.261  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
```

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

Nowadays, applications need several kinds of images as sources of information for elucidation and analysis. Several features are to be extracted so as to perform various applications. When an image is transformed from one form to another such as digitizing, scanning, and communicating, storing, etc. degradation occurs. Therefore ,the output image has to undertake a process called image enhancement, which contains of a group of methods that seek to develop the visual presence of an image. Image enhancement is fundamentally enlightening the interpretability or awareness of information in images for human listeners and providing better input for other automatic image processing systems. Image then undergoes feature extraction using various methods to make the image more readable by the computer. Sign language recognition system is a powerful tool to prepare an expert knowledge, edge detect and the combination of inaccurate information from different sources. \the intend of convolution neural network is to get the appropriate classification.

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 proposed sign language recognition system used to recognize sign language letters can be further extended to recognize gestures facial expressions. Instead of displaying letter labels it will be more appropriate to display sentences as more appropriate translation of language. This also increases read-ability. The scope of different sign languages can be increased. More training data can be added to detect the letter with more accuracy. This project can further be extended to convert the signs to speech.

The following are the features that can be added in 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 users' application.
- 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:

```
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=True)

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\IBM\RealtimeCommunicationSystemForSpeciallyAbled\Dataset\test_set\B\1.png") data=detect(frame)
```

Flask App Building:

```
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/C"
```

```
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)
```

Flash Application 3:

```
from flask import Flask, render_template,Response
from flask import Flask, Response, render_template
import cv2
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import numpy as np
import math

cap = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1)
classifier = Classifier("Model/keras_model.h5", "Model/labels.txt")

offset = 20
imgSize = 300

folder = "Data/C"
counter = 0

labels = ["A", "B", "C"]

while True:
success, img = cap.read()
```

```
imgOutput = img.copy()
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
        prediction, index = classifier.getPrediction(imgWhite, draw=False)
        print(prediction, index)

    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
        prediction, index = classifier.getPrediction(imgWhite, draw=False)
    cv2.rectangle(imgOutput, (x - offset, y - offset-50),
                  (x - offset+90, y - offset-50+50), (255, 0, 255), cv2.FILLED)
    cv2.putText(imgOutput, labels[index], (x, y -26), cv2.FONT_HERSHEY_COMPLEX, 1.7, (255, 255, 255), 2)
    cv2.rectangle(imgOutput, (x-offset, y-offset),
                  (x + w+offset, y + h+offset), (255, 0, 255), 4)



    cv2.imshow("ImageCrop", imgCrop)
    cv2.imshow("ImageWhite", imgWhite)

cv2.imshow("Image", imgOutput)
cv2.waitKey(1)
```

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>WebApp VideoTemplate</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="assets/css/Banner-Heading-Image.css">
    <link rel="stylesheet" href="assets/css/Navbar-Centered-Brand.css">
    <link rel="stylesheet" href="assets/css/styles.css">
</head>

<body style="background: rgb(255,255,255);">
    <nav class="navbar navbar-light navbar-expand-md py-3" style="background: #212529;">
        <div class="container">
            <div></div><a class="navbar-brand d-flex align-items-center" href="#"><span
```

```
                    class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex justify-content-center align-
items-center me-2 bs-icon"><i
                        class="fas fa-flask"></i></span><span style="color: rgb(255,255,255);">Real-Time
Communication
                    System Powered By AI&nbsp;For Specially Abled</span></a>
            <div></div>
        </div>
    </nav>
    <section>
        <div class="d-flex flex-column justify-content-center align-items-center">
            <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(0,0,0) ;">
                <img src="{{ url_for('video_feed') }}" style="width: 100%;height: 100%;color:
rgb(0,0,0);text-align: center;font-size: 20px;"
                    alt="Camera Hardware Can't Find!">
            </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> SL 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(0,0,0);">
                    <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(3,43,48);color: rgb(255,255,255);">Developed
By:</button></h2>
```

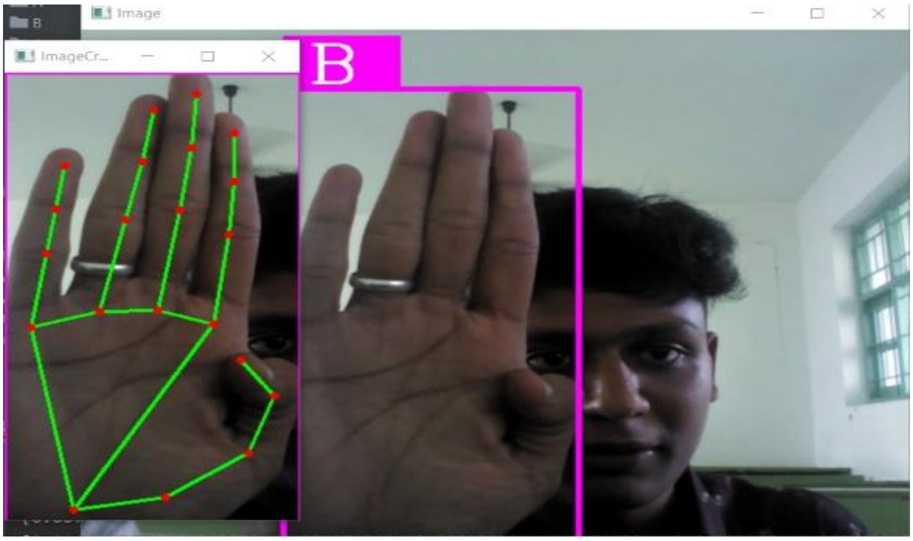
```
            <div class="accordion-body">
              <p>Team ID: PNT2022TMID05416
                <br><br>1. <strong> Balamurugan P </strong> 921319205014<br>2.
                <strong>Arunachalaa A S</strong> 921319205008<br>3. <strong>Arunpandian J</strong> 921319205011<br>4. <strong>Kailash A D</strong> 921319205052<br>
              </p>
            </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">
          <h5 class="modal-title">Sign Language - Alphabets</h5><button type="button"
            class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
        </div>
       <div class="modal-body">
        <div class="modal-footer"><button class="btn btn-secondary" type="button"
            data-bs-dismiss="modal">Close</button></div></div>
      </div>
    </div>
  </div>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>

</html>
```

Output



13.2 Github and Demo Link:

Github: https://github.com/IBM-EPBL/IBM-Project-9233-1658988732

Demo link:
https://drive.google.com/file/d/1ef1PyawQuHcKwgBXEfllyU_qAZyJLHx5/view?usp=share_link