**LOYOLA INSTITUTE OF TECHNOLOGY,CHENNAI**

**ANNA UNIVERSITY::CHENNAI – 600025**

**NALAIYA THIRAN PROJECT REPORT**

# REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABELED

**TEAM SIZE :4**

**TEAM ID: PNT2022TMID25716**

TEAM LEADER : GUHA IYISHVARYA V P

TEAM MEMBER 1: AMY DELPHIYA MARY R

TEAM MEMBER 2: UDHAYAGEETHA J

TEAM MEMBER 3: KEERTHANA V

# CONTENT

## 1. INTRODUCTION:

AI technologies can play an important role in breaking down the communication barriers of deaf or hearing-impaired people with other communities, contributing significantly to their social inclusion. Recent advances in both sensing technologies and AI algorithms have paved the way for the development of various applications aiming at fulfilling the needs of deaf and hearing-impaired communities. To this end, this survey aims to provide a comprehensive review of state-of-the-art methods in sign language capturing, recognition, translation and representation, pinpointing their advantages and limitations. In addition, the survey presents a number of applications, while it discusses the main challenges in the field of sign language technologies. Future research direction are also proposed in order to assist prospective researchers towards further advancing the field. Disability is an emotive human condition. It limits the individual to a certain level of performance. Being deaf and dumb pushes the subject to oblivion, highly introverted. In a tech era, no one should be limited due to his or her inability. The application of technology should create a platform or a world of equality despite the natural state of humans. On the other hand, technology is the most innovative thing on Earth for every time the clock ticks, researchers, software engineers, programmers, and information technology specialists are always coming up with bright ideas to provide convenience to everyone. This paper shows how artificial intelligence is being used to help people who are unable to do what most people do in their everyday lives. Aligned with communication, D-talk is a system that allows people who are unable to talk and hear be fully understood and for them to learn their language easier and also for the people that would interact and communicate with them. Natural Language Processing (NLP) is deals with the collaboration between computers and humans using the natural language. The ultimate objective of NLP is to read, decrypt, understand and make sense of the human languages in a manner that is valuable. Most NLP techniques rely on machine learning to derive meaning from human languages. AI is capable of deriving meaning from written and spoken words that enable us to better understand people, verify identities, anticipate world events, and solve crimes. CNN model to recognize gestures in sign language. Convolutional neural network of 11 layers is constructed, four Convolution layers, three Max-Pooling Layers, two dense layers, one flattening layer and one dropout layer. We use the Dataset to train the model to identify the gesture. The dataset contains the features of different augmented gestures.

## 1.1 Project Overview:

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Real-time communications (RTC) is any mode of telecommunications in which all users can exchange information instantly. Communication plays a significant role in making the world better place. It creates a bonding and relations among the people.

## 1.2 Purpose:

The main purpose of this review is to demonstrate the importance of using AI technologies in sign language to facilitate deaf and hearing-impaired people in their communication with other communities. 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:

### LITERATURE SURVEY- 1

**NAME:** A survey about deaf and dumb assisting system

**YEAR:** 2020

**DISCRIPTION:**

In this paper we tend to represent sensible forearm band for deaf and dumb patient. Concerning 9 billion individuals within the world are deaf and dumb. The communication between a deaf traditional visual individuals. This creates a awfully very little area for them with communication being a basic side of human life.

**ADVANTAGES:**

- The primary advantage is that the device can be carried out easily because of its low weight.

**DISADVANTAGES:**

- Easily exposed to noise Deployment of the environmental sensors based systems is limited to indoor environments.

## LITERATURE SURVEY-2

**NAME**: Accessibility to electronic communication for people with cognitive disabilities.

**YEAR:** 2019

**DISCRIPTION:**

The purpose of this study was to identify and synthesize measures for accessibility to electronic communication for people with cognitive disabilities by seeking answers to the following research questions: What measures to make electronic communication accessible to people with cognitive disabilities are evaluated and reported in the scientific literature? What documented effects do these measures have? Empirical studies describing and assessing cognitive accessibility measures were identified by searches of 13 databases.

**ADVANTAGES:**

- Low cost: Electronic communication saves time and money.

**DISADVANTAGES:**

- Humans cannot develop artificial intelligence because it is a technology based on pre-loaded facts and experience.

## LITERATURE SURVEY- 3

**NAME:** A survey on advanced technology for communication between deaf/dumb people using eye blink sensor & flex sensor.

**YEAR:** 2018

**ABSTRACT:**

The problem that our society faces nowadays is that people with disability are finding hard to grapple with the fast-growing technology. The access to communication technologies has become essential for the handicapped people. Normally deaf/dumb people use hand gestures for exchanging information and they find hard to deal with people who can't grasp the idea of sign language. And therefore they can't communicate with normal people.

**ADVANTAGES:**

- Some of the most technologically advanced companies engage with users using digital assistants, which eliminates the need for human personnel. Many websites utilize digital assistants to deliver user-requested content.

**DISADVANTAGES:**

- The ability to create a machine that can simulate human intelligence is no small feat.

## LITERATURE SURVEY-4

**NAME**: Hand gesture recognition based on deep learning

**Year:** 2017

**DISCRIPTION:**

This paper realizes the segmentation of hand gestures by establishing the skin color model and AdaBoost classifier based the according to the particularity of the skin for hand gestures, as well as the denaturation of hand gestures with one frame of video being cut for analysis. In this regard, the human hand is segmented from the complicated background, the real-time hand gestures tracking is also realized by Cam Shift algorithm.

**ADVANTAGES:**

- It helps in understanding the people of deaf and dumb.

**DISADVANTAGE:**

- Accuracy of hand gesture recognition may be varying due to no-standard background.

## 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:

- Dhanraj K.V.,Kriplani L. and Mahajan S., ‘Research Paper on Desktop Voice Assistant’(2022),International Journal of Research in Engineering and Science (IJRES),Vol 10,Issue 2
- Damarla K.,‘Virtual Assistant Using Python’, Retrieved from https://extrudesign.com/virtualassistant-using-python/
- Empowering people with disabilities through AIhttps://blogs.microsoft.com/uploads/2018/02/TheFuture-Computed_2.8.18.pdf (accessed Sep. 26, 2021)

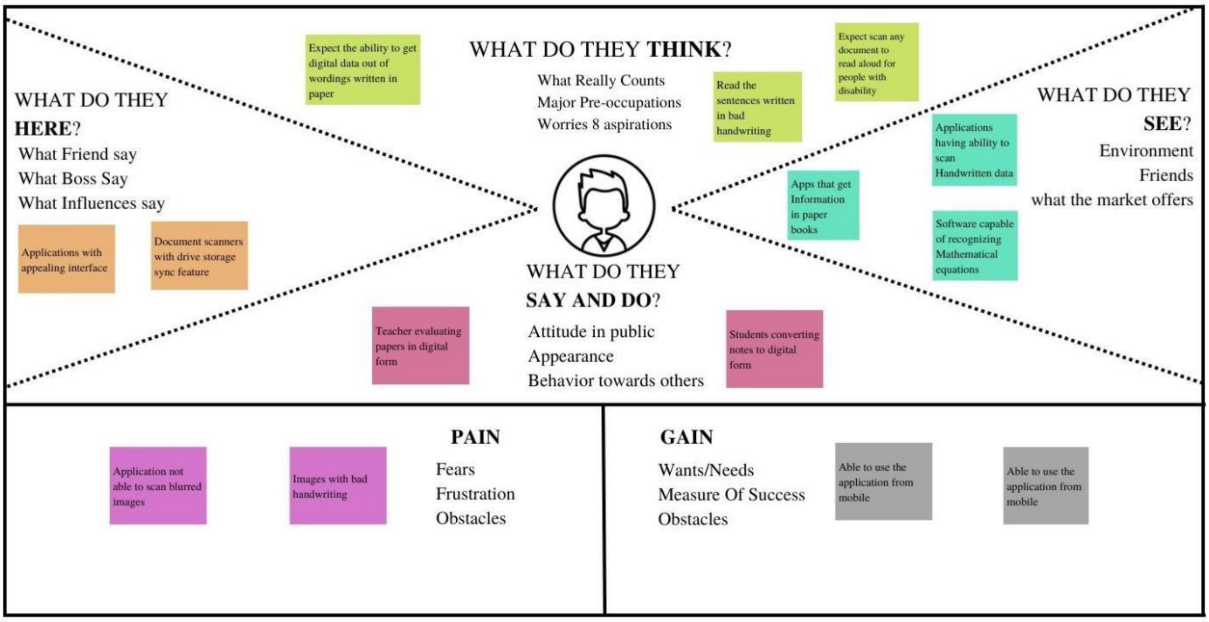
## 2.3 Problem Solution Definition:

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.

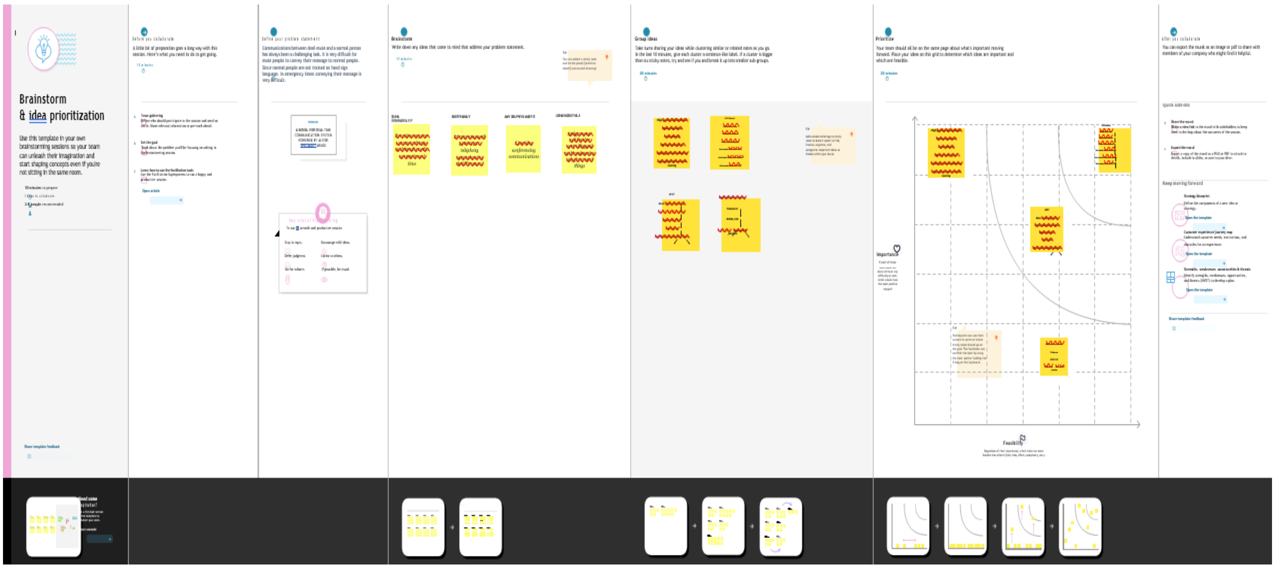
# 3.IDEATION &PROPOSED SOLUTION

## 3.1 Empathy Map Canvas:

EMPATHY MAP


WHAT DO THEY HERE?
What Friend say
What Boss Say
What Influences say
Applications with appealing interface
Document scanners with drive storage sync feature
Expect the ability to get digital data out of wordings written in paper
WHAT DO THEY THINK?
What Really Counts
Major Pre-occupations
Worries 8 aspirations
Read the sentences written in bad handwriting
Expect scan any document to read aloud for people with disability
WHAT DO THEY SEE?
Environment
Friends
what the market offers
Applications having ability to scan Handwritten data
Apps that get Information in paper books
Software capable of recognizing Mathematical equations
WHAT DO THEY SAY AND DO?
Attitude in public
Appearance
Behavior towards others
Teacher evaluating papers in digital form
Students converting notes to digital form
PAIN
Fears
Frustration
Obstacles
Application not able to scan blurred images
Images with bad handwriting
GAIN
Wants/Needs
Measure Of Success
Obstacles
Able to use the application from mobile
Able to use the application from mobile

# 3.2 Ideation & 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.

10 minutes to prepare

1 hour to collaborate

2-8 people recommended

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.

10 minutes

Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

Open article

Define your problem statement

Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult.

A NOVEL FOR REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

Encourage wild ideas.

Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.

Brainstorm

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

10 minutes

Tip: You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural

Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural

Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint

Define the components of a new idea or strategy.

Open the template

Customer experience journey map

Understand customer needs, motivations, and obstacles for an experience.

Open the template

Strengths, weaknesses, opportunities & threats

Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

Open the template

Share template feedback

## 3.3 Proposed Solution:

### 1.Problem Statement (Problem to be solved) :

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

### 2.Idea / Solution description:

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

### 3. Social Impact / Customer Satisfaction :

The proposed solution is keen on providing a friendly user interface and user experience. User Interface (UI) is aimed to be developed in such a that way that it can be very handy and easy to learn. The system is also aimed to be light weight which would make the system provide faster and accurate results and hence it provides a better User Experience (UX).

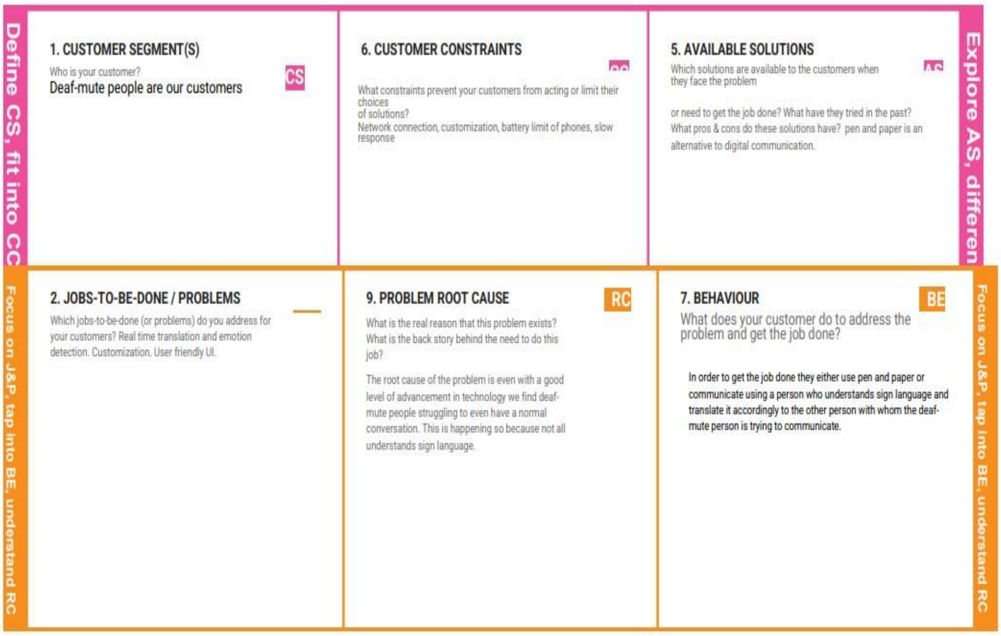
### 4.Business Model (Revenue Model):

The proposed solutions help to ease the communication between deaf and dumb people and normal people. The customization and emotion detection feature can make it lot more reliable. Hence, the solution has wide usability and requirement.

### 5.Scalability of the Solution:

This proposed solution is highly extensible in terms of the features that is been offered by the system. It can be seen as a highly improvised and light weight model when compared to the existing systems. The system can further be scaled in such a way that enables tasks being assigned and completed in system through gestures.

# 3.4 Problem solution fit:



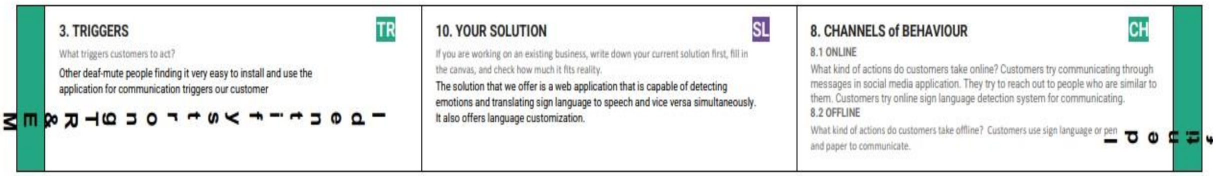
Define CS, fit into CC

Explore AS, differen

|  |  |  |
|---|---|---|
| 1. CUSTOMER SEGMENT(S)<br>CS<br>Who is your customer?<br>Deaf-mute people are our customers | 6. CUSTOMER CONSTRAINTS<br>CC<br>What constraints prevent your customers from acting or limit their choices of solutions?<br>Network connection, customization, battery limit of phones, slow response | 5. AVAILABLE SOLUTIONS<br>AS<br>Which solutions are available to the customers when they face the problem<br>or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? pen and paper is an alternative to digital communication. |
| 2. JOBS-TO-BE-DONE / PROBLEMS<br>Which jobs-to-be-done (or problems) do you address for your customers? Real time translation and emotion detection. Customization. User friendly UI. | 9. PROBLEM ROOT CAUSE<br>RC<br>What is the real reason that this problem exists? What is the back story behind the need to do this job?<br>The root cause of the problem is even with a good level of advancement in technology we find deaf-mute people struggling to even have a normal conversation. This is happening so because not all understands sign language. | 7. BEHAVIOUR<br>BE<br>What does your customer do to address the problem and get the job done?<br>In order to get the job done they either use pen and paper or communicate using a person who understands sign language and translate it accordingly to the other person with whom the deaf-mute person is trying to communicate. |

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

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



|  |  |  |
|---|---|---|
| 3. TRIGGERS<br>TR<br>What triggers customers to act?<br>Other deaf-mute people finding it very easy to install and use the application for communication triggers our customer<br>Identify strong TR & EM | 10. YOUR SOLUTION<br>SL<br>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.<br>The solution that we offer is a web application that is capable of detecting emotions and translating sign language to speech and vice versa simultaneously. It also offers language customization. | 8. CHANNELS of BEHAVIOUR<br>CH<br>8.1 ONLINE<br>What kind of actions do customers take online? Customers try communicating through messages in social media application. They try to reach out to people who are similar to them. Customers try online sign language detection system for communicating.<br>8.2 OFFLINE<br>What kind of actions do customers take offline? Customers use sign language or pen and paper to communicate.<br>Identif |

# 4. REQUIREMENT ANALYSIS:

## 4.1 Functional requirements:

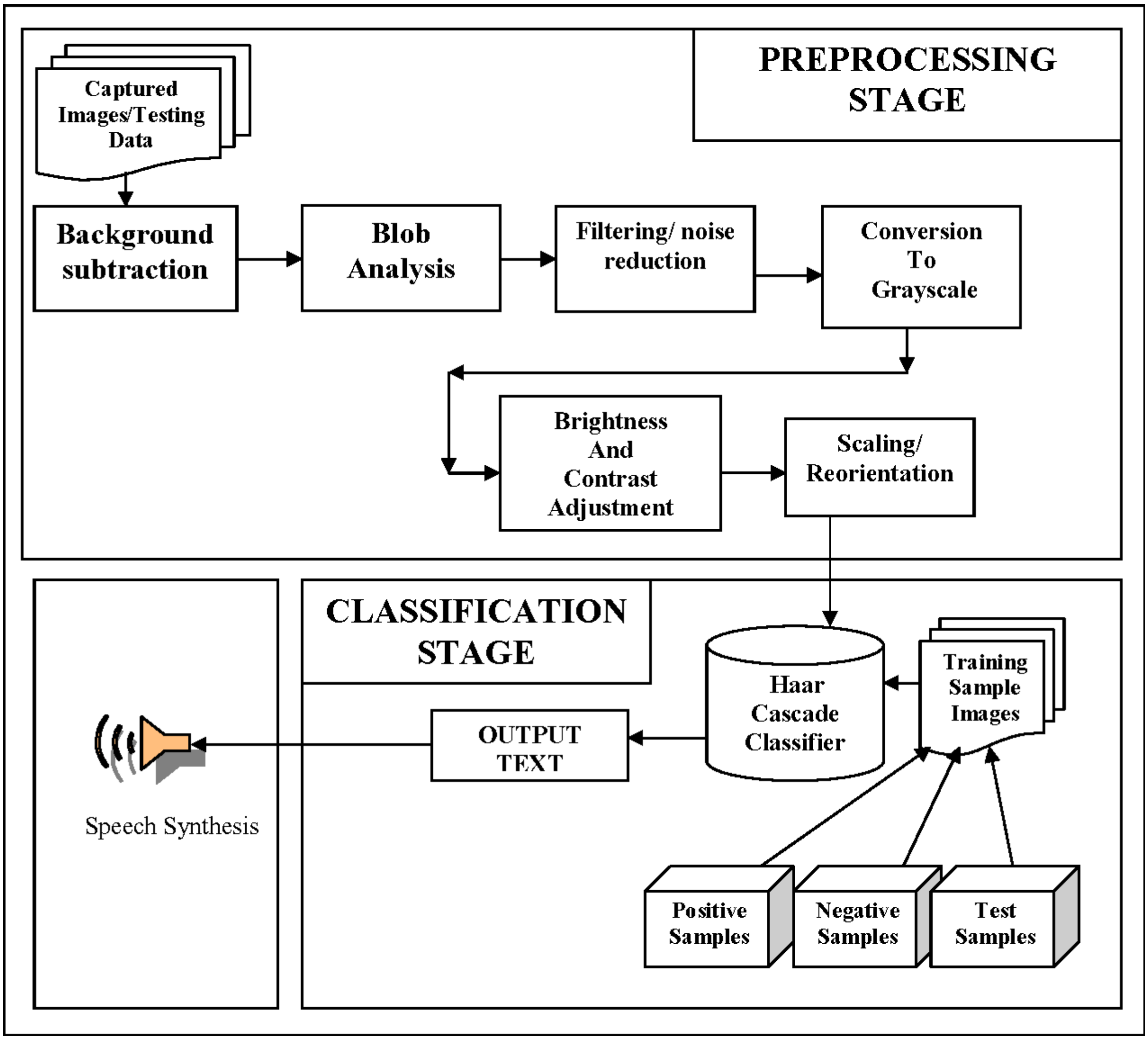
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
|---|---|---|
| FR-1 | **Language customization** | The user performs language customization. |
| FR-2 | **User Options** | The user either chooses to convert speech to sign language and sign language to speech. |
| FR-3 | **Test Inputs** | The real time video and audio data is collected and fed into the machine learning model. |
| FR-4 x | **Result** | The conversion will take place simultaneously and will be displayed on the screen. |

## 4.2 Non-Functional Requirements:

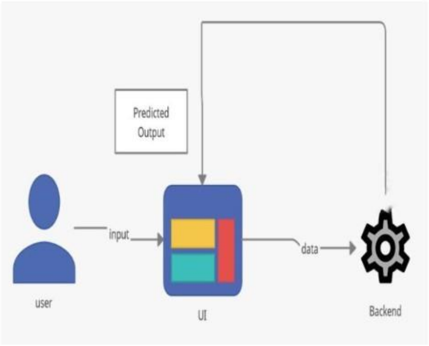
| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | The user will have access to all the resources present in the website. |
| NFR-2 | **Security** | User information is protected. |
| NFR-3 | **Reliability** | It offers accurate results. |
| NFR-4 | **Performance** | The web application makes use of light weight model hence the result will be accurate and fast. |
| NFR-5 | **Availability** | The web application can be accessed 24/7 from anywhere when connected to the internet. |
| NFR-6 | **Scalability** | The trained ML model can provide accurate results whenever the size of the dataset and the number of users is extended. |

# 5. PROJECT DESIGN:

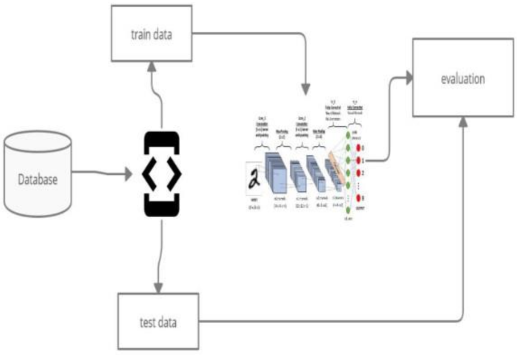
## 5.1 Data Flow Diagram:



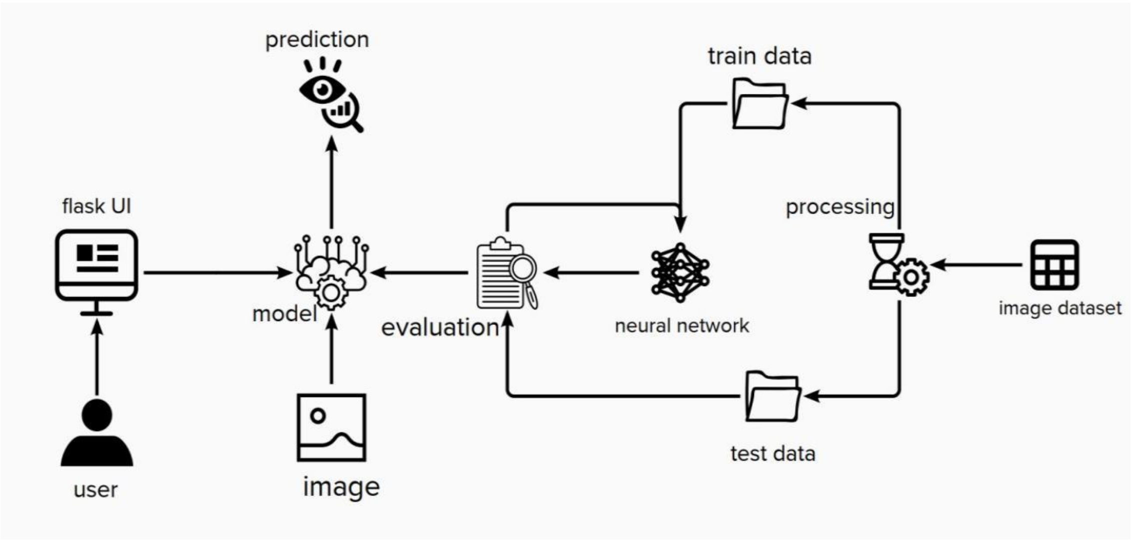
## 5.2 Solution and Technical Architecture:



DEPLOYMENT



TRAINING AND EVALUATION



## 5.3 **User stories:**

| **User type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story/Task** | **criteria** | **Priority** | **Release** |
|---|---|---|---|---|---|---|
| Customer | Uploading the real time data. | USN-1 | The user will be presented with two options.<br>Speech to sign language conversion.<br>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 processi ng | High | Sprint-1 |

# 6. PROJECT PLANNING & SCHEDULING:

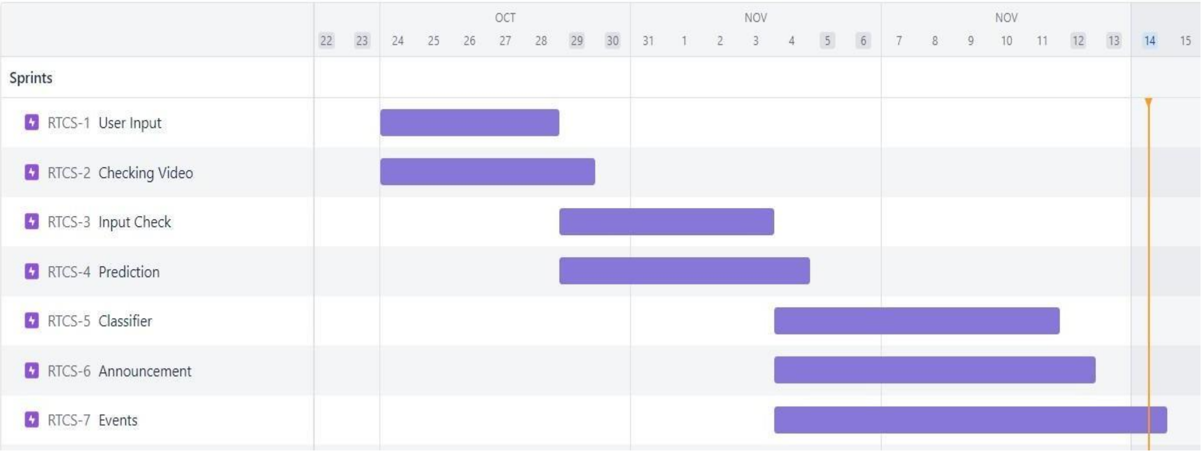
## 6.1 Sprint planning & Estimation:

| SPRINT | FUNCTIONAL REQUIREMENTS | USER STORY NUMBER | USER STORY/TASK | STORY POINTS | PRIORITY | TEAM MEMBERS |
|---|---|---|---|---|---|---|
| Sprint -1 | Registration | USN - 1 | Collect Dataset | 2 | High | Guha iyishvarya V PAmy delphiya mary R Keerthana V Udhayageetha J |
| Sprint -1 | Login | USN - 2 | Collectin gkey points using Media Pipe Holistic | 1 | High | Guha iyishvarya V PAmy delphiya mary R |
| Sprint -2 |  | USN - 3 | Model Initialisation With required layers | 1 | Medium | Keerthana V Udhayageetha J |
| Sprint -2 | Dashboard | USN - 4 | As a user, I can log intomy account in a given Dashboard | 1 | High | Guha iyishvarya V P Amy delphiya mary R |
| Sprint -1 | User Interface | USN – 5 | Professional responsible for user Requirements & needs | 1 | High | Keerthana V Udhayageetha J |
| Sprint -3 | Objective | USN - 6 | The goal is to describe all the inputs & Outputs | 1 | High | Amy delphiya mary R Udhayageetha J |
| Sprint -4 | Privacy | USN - 7 | The Developed application should be secure forthe Users. | 1 | High | Keerthana V Guha iyishvarya V P |

## 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 |
|---|---|---|---|---|---|---|
| Sprint- 1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint- 2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint- 3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint- 4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

## 6.3 Report From JIRA:



## 7. CODING & SOLUTION :

### 7.1. Feature 1:

We added numbers and alphabets hand signs as image inputs using hand signs

The below code is used for this feature

**Main code**

```
    from tensorflow.keras.preprocessing.image

    import ImageDataGenerator from

    tensorflow.keras.models import Sequential

    from tensorflow.keras.layers import
    Convolution2D,MaxPooling2D,Flatten,Dense

    train_data=ImageDataGenerator(rescale=1./255,zoom_ran
    ge=0.2,horizontal_flip=Tru
e)

    test_data=ImageDataGenerator(rescale=1./255,validatio

    n_split=0.5)

    xtrain=train_data.flow_from_directory('E:/Projects/Jyupt

    er/Dataset/Hand Sign/Train',

    target_size=(64,64),class_mode='categorical',batch_size

    =100)

    xtest=test_data.flow_from_directory('E:/Projects/Jyupter

    /Dataset/Hand Sign/Test',
        target_size=(64,64),class_mode='categorical',batch_size=100)

        #Layering
        model=Sequential()
        model.add(Convolution2D(32,(3,3),activation='relu',input_shape
        =(64,64,3))) model.add(MaxPooling2D(pool_size=(2,2)))
        model.add(Flatten())
        model.add(Dense(300,activation='rel
        u'))
        model.add(Dense(150,activation='rel
        u'))
        model.add(Dense(36,activation='soft
        max'))

        #Compile
        model.compile(optimizer='adam',loss='categorical_crossentropy',
        metrics=['accuracy']
)
#Fit
  Model.fit(xtrain,steps_per_epoch=len

  (xtinra),
model.fit(xtrain,steps_per_epoch=len(xtrain
),
        epochs=25,validation_data=xtest,validation_ste
        ps=len(xtest)) model.save('Hand-SignV2.h5')

        #Test
        from tensorflow.keras.preprocessing
        import image import numpy as np
        fl_img='E:/Projects/Jyupter/Dataset/H
        and
        Sign/Test/9/hand1_9_left_seg_4_crop
        ped.jpeg'
        img=image.load_img(fl_img,target_size
        =(64,64)) x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        pred=np.argmax(model.predict(x))
        print(pred)
```

```
op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','
K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

op[pred]

fl_img='E:/Projects/Jyupter/Dataset/conversation
engine for deaf and dumb/Dataset/test_set/A/15.png'

img=image.load_img(fl_img,target_size

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

x=np.expand_dims(x,axis=0)

pred=np.argmax(model.predict(x))

print(pred)

op=['A','B','C','D','E','F','G','

H','I']

op[pred]

import tensorflow.keras.models

new_model = tensorflow.keras.models.load_model('Hand-
SignV2.h5')

fl_img='E:/Projects/Jyupter/Dataset/conversation
engine for deaf and dumb/Dataset/test_set/A/15.png'

img=image.load_img(fl_img,target_size

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

x=np.expand_dims(x,axis=0)

pred=np.argmax(new_model.predict(x))

print(pred)

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

```
op[pred]
fl_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1_9_left_seg_4_cropped.jpeg'
img=image.load_img(fl_img,target_size
=(64,64)) x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(new_model.predict(x))
print(pred)
op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L ','M','N','O ','P','Q','R','S','T','U','V','W','X','Y','Z']
op[pred]
```

app.py

```
#Import necessary libraries
from flask import Flask, render_template,
Response,url_for import cv2
from keras.preprocessing import image
from keras.preprocessing.image import
load_img import numpy as np
from keras.models import load_model
```

```
     #Initialize the
Flask app
app= Flask(_name_)
@app.route('/')
        def index():
          return
        render_template('index.ht
        ml')
        @app.route('/video_feed'
        )
        def video_feed():
          new_model =
          load_model('Hand-
          SignV2.h5')cap =


          cv2.VideoCapture(0)
          try:
            while True:
              cap.set(cv2.CAP_PROP_POS_M
              SEC,1000*1000)ret, frame =
              cap.read()
```

```
       frame =

       v2.flip(frame, 1)

       cv2.imshow('Inpu

       t', frame) if ret:
        cv2.imwrite("image.jpeg",frame)
        img = image.load_img("./image.jpeg",
        target_size=(64, 64))


        x = image.img_to_array(img)


        x = np.expand_dims(x, axis=0)
        pred = np.argmax(new_model.predict(x))
        op = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C',
        'D', 'E', 'F', 'G', 'H', 'I', 'J',
    'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y',
    'Z']

       cv2.waitKe

       y(250) yield

       op[pred]

     except

      KeyboardInterr

      upt: return -1



if _name=='_main_': app.run(debug=True) new1.css
```

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wg
ht@
400;500;600&display= swap');
body{
background: #FC7307;
}
a{
  color: blue;
  text-align:
  center;
  display:
  block; font-
  size:1.8em;
}
.two{
 margin-left:
 auto;
 margin-
 right: auto;
}
h1{
color : #236AB9;
}
h2{
  color : #341C09;
}
```

**7.2. Feature 2:**

We used our webcam live feed as input. So when the user shows a hand sign the CNN predicts the correct alphabet or number and prints as a text

The below code is used for this feature feed.html

```
<html>
  <body style="color:#D4E4F7 ">
  {{ cv2.capture(0) }}
  <br>
  {% for i in video_feed() %}
    <h4>{{ i }}</h4>
  {% endfor %}
  </body>
</html>
```

index.html

```
<html>
  <head>
    <link rel= "stylesheet" type=
"text/css" href= "{{
url_for('static',filename='new1.css')
}}">
```

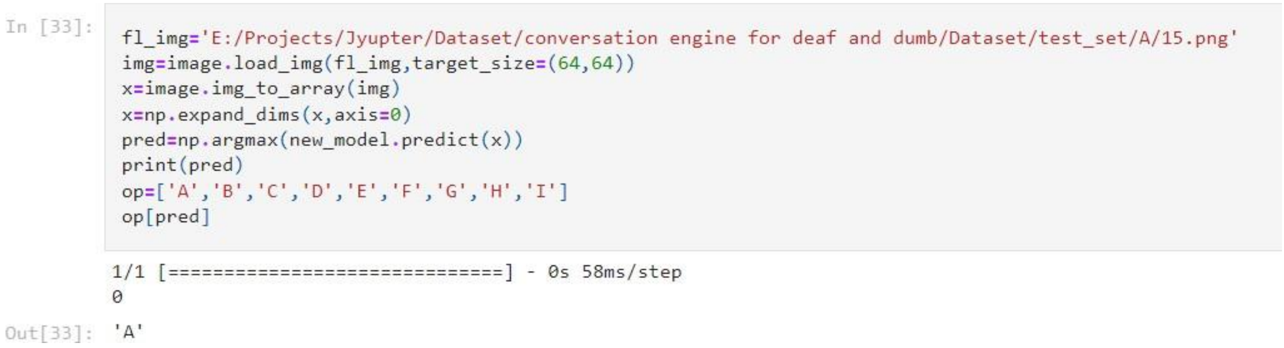
```
    Video Streaming Live Web Cam
  </head>

  <title>Real-Time Communication System
Powered by AI for Specially Abled</title>
  <style>
      h1 {text-align:
   center;} h2 {text-
   align: center;}
   a {text-align: center;}
  </style>
  <body>
          <div
    class="containe
          r">
          <div
     class="row">
        <h1 style="font-size:45px">Real-Time
Communication System Powered by AI for Specially
Abled</h1>
        <h2 style="font-size:35px">Live Streaming</h3>
        <a href="{{ url_for('video_feed') }}" class="btn
        btn-warning">Start</a>
      </div>
  </div>
    <img class="one" src="{{ url_for('static',
      filename='hand-sign.jpg') }}"width="1000"
  height="400">
</body>
 </html>
```

# 8. TESTING :

## 8.1 Test Cases:

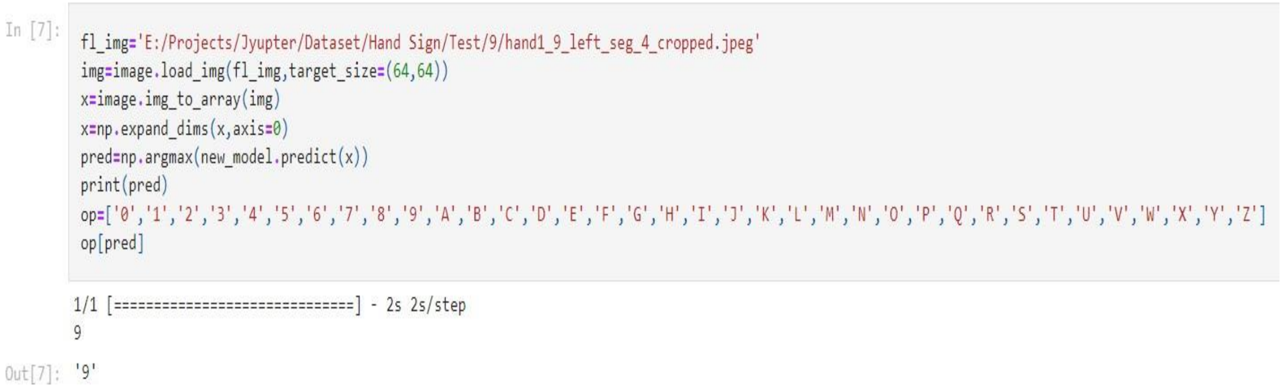
### Test Case 1



```
In [33]:  fl_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test_set/A/15.png'
          img=image.load_img(fl_img,target_size=(64,64))
          x=image.img_to_array(img)
          x=np.expand_dims(x,axis=0)
          pred=np.argmax(new_model.predict(x))
          print(pred)
          op=['A','B','C','D','E','F','G','H','I']
          op[pred]

          1/1 [==============================] - 0s 58ms/step
          0
Out[33]:  'A'
```

### Test Case 2



```
In [7]:  fl_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1_9_left_seg_4_cropped.jpeg'
         img=image.load_img(fl_img,target_size=(64,64))
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         pred=np.argmax(new_model.predict(x))
         print(pred)
         op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
         op[pred]

         1/1 [==============================] - 2s 2s/step
         9
Out[7]:  '9'
```

## 8.2 User acceptance Testing:

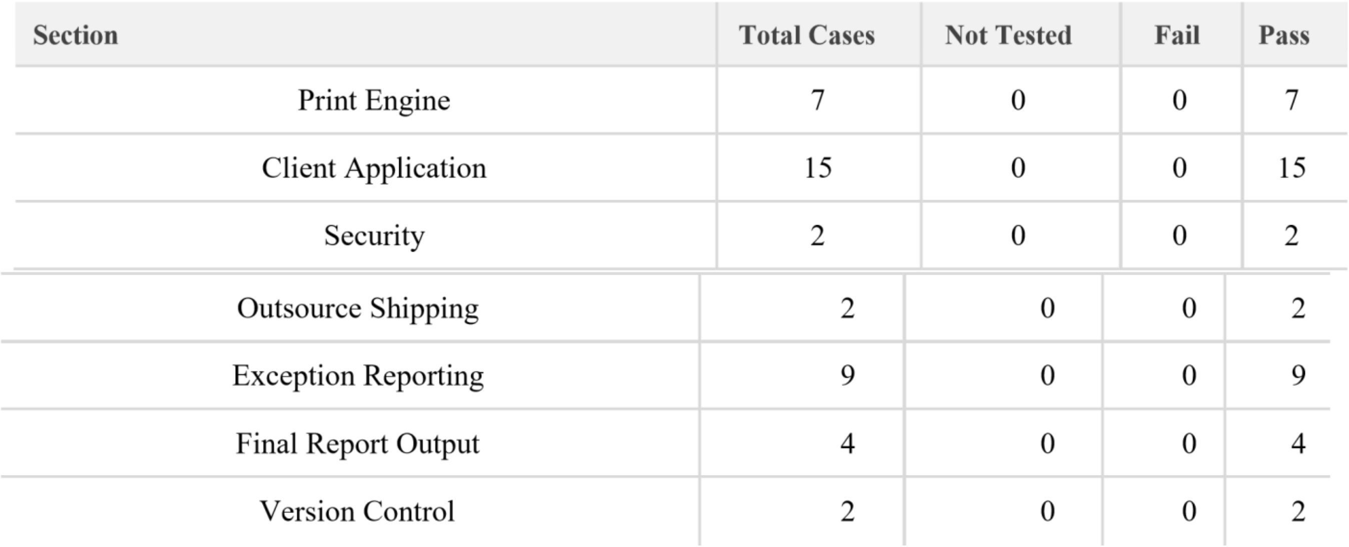
### 1. Defect Analysis:

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

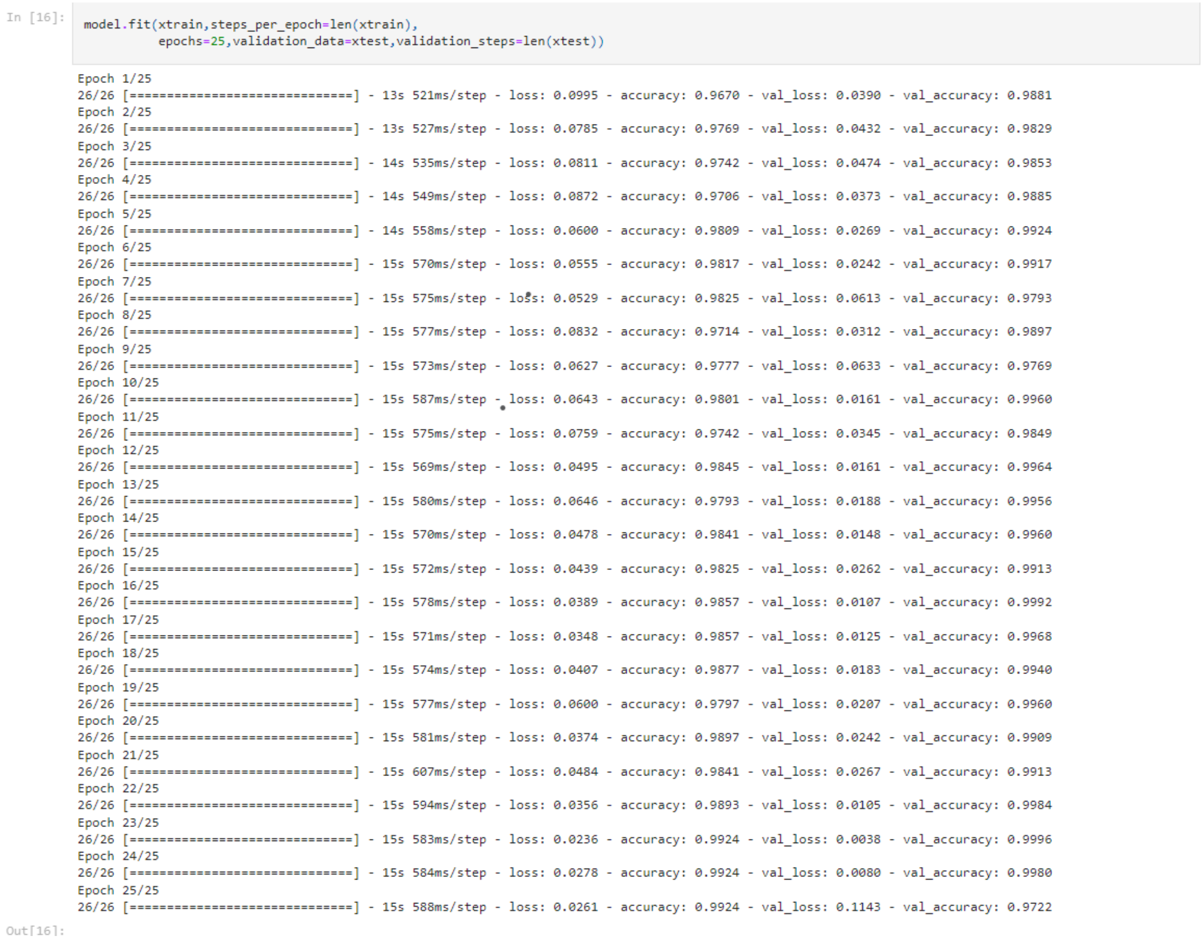


| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| Print Engine | 7 | 0 | 0 | 7 |
| Client Application | 15 | 0 | 0 | 15 |
| Security | 2 | 0 | 0 | 2 |
| Outsource Shipping | 2 | 0 | 0 | 2 |
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version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

# 9. RESULT:

## 9.1 Performance Metrics:

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



```
In [16]: model.fit(xtrain,steps_per_epoch=len(xtrain),
                  epochs=25,validation_data=xtest,validation_steps=len(xtest))

         Epoch 1/25
         26/26 [==============================] - 13s 521ms/step - loss: 0.0995 - accuracy: 0.9670 - val_loss: 0.0390 - val_accuracy: 0.9881
         Epoch 2/25
         26/26 [==============================] - 13s 527ms/step - loss: 0.0785 - accuracy: 0.9769 - val_loss: 0.0432 - val_accuracy: 0.9829
         Epoch 3/25
         26/26 [==============================] - 14s 535ms/step - loss: 0.0811 - accuracy: 0.9742 - val_loss: 0.0474 - val_accuracy: 0.9853
         Epoch 4/25
         26/26 [==============================] - 14s 549ms/step - loss: 0.0872 - accuracy: 0.9706 - val_loss: 0.0373 - val_accuracy: 0.9885
         Epoch 5/25
         26/26 [==============================] - 14s 558ms/step - loss: 0.0600 - accuracy: 0.9809 - val_loss: 0.0269 - val_accuracy: 0.9924
         Epoch 6/25
         26/26 [==============================] - 15s 570ms/step - loss: 0.0555 - accuracy: 0.9817 - val_loss: 0.0242 - val_accuracy: 0.9917
         Epoch 7/25
         26/26 [==============================] - 15s 575ms/step - loss: 0.0529 - accuracy: 0.9825 - val_loss: 0.0613 - val_accuracy: 0.9793
         Epoch 8/25
         26/26 [==============================] - 15s 577ms/step - loss: 0.0832 - accuracy: 0.9714 - val_loss: 0.0312 - val_accuracy: 0.9897
         Epoch 9/25
         26/26 [==============================] - 15s 573ms/step - loss: 0.0627 - accuracy: 0.9777 - val_loss: 0.0633 - val_accuracy: 0.9769
         Epoch 10/25
         26/26 [==============================] - 15s 587ms/step - loss: 0.0643 - accuracy: 0.9801 - val_loss: 0.0161 - val_accuracy: 0.9960
         Epoch 11/25
         26/26 [==============================] - 15s 575ms/step - loss: 0.0759 - accuracy: 0.9742 - val_loss: 0.0345 - val_accuracy: 0.9849
         Epoch 12/25
         26/26 [==============================] - 15s 569ms/step - loss: 0.0495 - accuracy: 0.9845 - val_loss: 0.0161 - val_accuracy: 0.9964
         Epoch 13/25
         26/26 [==============================] - 15s 580ms/step - loss: 0.0646 - accuracy: 0.9793 - val_loss: 0.0188 - val_accuracy: 0.9956
         Epoch 14/25
         26/26 [==============================] - 15s 570ms/step - loss: 0.0478 - accuracy: 0.9841 - val_loss: 0.0148 - val_accuracy: 0.9960
         Epoch 15/25
         26/26 [==============================] - 15s 572ms/step - loss: 0.0439 - accuracy: 0.9825 - val_loss: 0.0262 - val_accuracy: 0.9913
         Epoch 16/25
         26/26 [==============================] - 15s 578ms/step - loss: 0.0389 - accuracy: 0.9857 - val_loss: 0.0107 - val_accuracy: 0.9992
         Epoch 17/25
         26/26 [==============================] - 15s 571ms/step - loss: 0.0348 - accuracy: 0.9857 - val_loss: 0.0125 - val_accuracy: 0.9968
         Epoch 18/25
         26/26 [==============================] - 15s 574ms/step - loss: 0.0407 - accuracy: 0.9877 - val_loss: 0.0183 - val_accuracy: 0.9940
         Epoch 19/25
         26/26 [==============================] - 15s 577ms/step - loss: 0.0600 - accuracy: 0.9797 - val_loss: 0.0207 - val_accuracy: 0.9960
         Epoch 20/25
         26/26 [==============================] - 15s 581ms/step - loss: 0.0374 - accuracy: 0.9897 - val_loss: 0.0242 - val_accuracy: 0.9909
         Epoch 21/25
         26/26 [==============================] - 15s 607ms/step - loss: 0.0484 - accuracy: 0.9841 - val_loss: 0.0267 - val_accuracy: 0.9913
         Epoch 22/25
         26/26 [==============================] - 15s 594ms/step - loss: 0.0356 - accuracy: 0.9893 - val_loss: 0.0105 - val_accuracy: 0.9984
         Epoch 23/25
         26/26 [==============================] - 15s 583ms/step - loss: 0.0236 - accuracy: 0.9924 - val_loss: 0.0038 - val_accuracy: 0.9996
         Epoch 24/25
         26/26 [==============================] - 15s 584ms/step - loss: 0.0278 - accuracy: 0.9924 - val_loss: 0.0080 - val_accuracy: 0.9980
         Epoch 25/25
         26/26 [==============================] - 15s 588ms/step - loss: 0.0261 - accuracy: 0.9924 - val_loss: 0.1143 - val_accuracy: 0.9722
Out[16]:
```

## 10. ADVANTAGES & DISADVANTAGES:

### Advantages:

- Real time sign to speech detection & Real time facial emotion detection.
- Model provides good accuracy.
- Real time facial emotion detection & Real time speech to text conversion.
- Data privacy.
- Before starting the video feed we provide an image which shows what hand sign corresponds to what alphabet , In other words our model is user friendly.

### Disadvantages:

- Our model is not 100% effective,some flaws in detecting the hand sign may occur.Its mainly because each person use different angles while using hand signs so in detecting the hand sign based on a specific set of data sets is not sufficient.
- User cannot make calls using the app.

## 11. CONCLUSION:

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

## 12.FUTURE SCOPE:

- 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:

Main code

```
        from tensorflow.keras.preprocessing.image

        import ImageDataGenerator from

        tensorflow.keras.models import Sequential

        from tensorflow.keras.layers import
        Convolution2D,MaxPooling2D,Flatten,Dense

        train_data=ImageDataGenerator(rescale=1./255,zoom_ran
        ge=0.2,horizontal_flip=Tru
  e)
  test_data=ImageDataGenerator(rescale=1./255,validation_split
  =0.5)
  xtrain=train_data.flow_from_directory('E:/Projects/Jyupter/Dat
  aset/Hand Sign/Train',
  target_size=(64,64),class_mode='categorical',batch_size=100)
  xtest=test_data.flow_from_directory('E:/Projects/Jyupter/Datas
  et/Hand Sign/Test',

  target_size=(64,64),class_mode='categorical',batch_size=100


# Layering

model=Sequential()

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

```
    model.add(MaxPooling2D(pool_size=(2,2)))

    model.add(Flatten())

    model.add(Dense(300,activati

    on='relu'))

    model.add(Dense(150,activati

    on='relu'))

    model.add(Dense(36,activatio

    n='softmax'))


    #Compile
    model.compile(optimizer='adam',loss='categorical_crosse
    ntropy',metrics=['accuracy']
)


    #Fit

    model.fit(xtrain,steps_per_ep

    och=len(xtrain),

    epochs=25,validation_data=xtest,validati

    on_steps=len(xtest)) model.save('Hand-

    SignV2.h5')
```

```
#Test

From

tensorflow.keras.preprocessing

import image import numpy as np

fl_img='E:/Projects/Jyupter/Da
taset/Hand
Sign/Test/9/hand1_9_left_seg_
4_cropped.jpeg'

img=image.load_img(fl_img,targ

et_size=(64,64))

x=image.img_to_array(img)

x=np.expand_dims(x,axis=0)

pred=np.argmax(model.predict(x

))
print(pred)

op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H
','I','J','K','L','M','N','O','P','Q',
'R','S','T','U','V','W','X','Y','Z']


op[pred]

fl_img='E:/Projects/Jyupter/Dataset/conversat
ion engine for deaf and
dumb/Dataset/test_set/A/15.png'
```

```
img=image.load_img(fl_img,targ
et_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(model.predict(x
))
print(pred)
op=['A','B','C','D','E','
F','G','H','I']
op[pred]
import tensorflow.keras.models
new_model =
tensorflow.keras.models.load_model('Hand-SignV2.h5')
fl_img='E:/Projects/Jyupter/Dataset/conversat
ion engine for deaf and
dumb/Dataset/test_set/A/15.png'
img=image.load_img(fl_img,targ
et_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(new_model.pre
dict(x)) print(pred)
op=['A','B','C','D','E','F','G','H','I']
op[pred]
```

```
fl_img='E:/Projects/Jyupter/Da
taset/Hand
Sign/Test/9/hand1_9_left_seg_
4_cropped.jpeg'

img=image.load_img(fl_img,targ

et_size=(64,64))

x=image.img_to_array(img)

x=np.expand_dims(x,axis=0)

pred=np.argmax(new_model.pre

dict(x)) print(pred)

op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H
','I','J','K','L','M','N','O','P','Q',
'R','S','T','U','V','W','X','Y','Z']



op[pred]
```

app.py

```
#Import necessary libraries

from flask import Flask,

render_template, Response,url_for

import cv2

from keras.preprocessing import image

from keras.preprocessing.image

import load_img import numpy
```

```
from keras.preprocessing.image
import load_img import numpy
as np

from keras.models import load_model

#Initialize the
Flask app app =
Flask(_name_)
@app.route('/')
def index():
  return
render_template('index.ht
ml')
@app.route('/video_feed'
)
def video_feed():
  new_model =
  load_model('Hand-
  SignV2.h5') cap =
  cv2.VideoCapture(0)
```

```
try:
        while True:
          cap.set(cv2.CAP_PROP_POS_M
          SEC,1000*1000)ret, frame =
          cap.read()
          frame =
          cv2.flip(frame, 1)
          cv2.imshow('Inpu
          t', frame)if ret:
            cv2.imwrite("image.jpeg",frame)
            img = image.load_img("./image.jpeg",
            target_size=(64, 64))x =
            image.img_to_array(img)


            x = np.expand_dims(x, axis=0)
            pred = np.argmax(new_model.predict(x))
            op = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C',
            'D', 'E', 'F', 'G', 'H', 'I', 'J',
      'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y',
      'Z']
```

```
cv2.waitKey(250) yield
op[pred]
            except
              KeyboardInterr
              upt: return -1
          if _name=='
            main_':
            app.run(debu
            g=True)
```

new1.css

```
          @import
          url('https://fonts.googleapis.com/css2?family=Poppins:wg
          ht@400;500;600&display= swap');
          body{
            background: #FC7307;
          }
          a{
            color: blue;
            text-align:
            center;
            display:
            block; font-
            size:1.8em;
          }
          .two{
           margin-left:
           auto;
           margin-
           right: auto;
          }

          h1{

          color : #236AB9;
          }
          h2{
            color : #341C09;
          }
```

feed.html

```
<html>
            <body style="color:#D4E4F7 ">
            {{ cv2.capture(0) }}
            <br>
```

```
  {% for i in video_feed() %}
    <h4>{{ i }}</h4>
  {% endfor %}
  </body>
</html>
```

index.html

```
<html>
  <head>
    <link rel= "stylesheet" type=
"text/css" href= "{{
url_for('static',filename='new1.css')
}}">


    Video Streaming Live Web Cam
  </head>


  <title>Real-Time Communication System
Powered by AI for Specially Abled</title>
  <style>
      h1 {text-align:
   center;} h2 {text-
   align: center;}
```

```
     a {text-align: center;}
    </style>
    <body>
              <div
       class="containe
               r">
              <div
        class="row">
          <h1 style="font-size:45px">Real-Time
  Communication System Powered by AI for Specially
  Abled</h1>
          <h2 style="font-size:35px">Live Streaming</h3>
          <a href="{{ url_for('video_feed') }}" class="btn
          btn-warning">Start</a>
        </div>
   </div>
      <img class="one" src="{{ url_for('static',
  filename='hand-sign.jpg') }}"width="1000"
  height="400">
 </body>
  </html>
```

Project_Default.xml

```
<component name="InspectionProjectProfileManager">
 <profile version="1.0">
  <option name="myName" value="Project Default" />
  <inspection_tool class="PyPep8Inspection"
enabled="true" level="WEAK WARNING"
```

```
enabled_by_default="true">
   <option name="ignoredErrors">
    <list>
     <option value="E302" />
     <option value="E305" />
    </list>
   </option>
  </inspection_tool>
  <inspection_tool class="PyPep8NamingInspection"
enabled="true" level="WEAK WARNING"
enabled_by_default="true">
   <option name="ignoredErrors">
    <list>
     <option value="N801" />
    </list>
   </option>
  </inspection_tool>
 </profile>



</component>
```

profiles_settings.xml

```
<component name="InspectionProjectProfileManager">
 <settings>
  <option name="USE_PROJECT_PROFILE"
  value="false" />
  <version value="1.0" />
 </settings>
</component>
```

IBM.iml

```
<?xml version="1.0" encoding="UTF-8"?>
<module type="PYTHON_MODULE" version="4">
 <component name="NewModuleRootManager">
  <content url="file://$MODULE_DIR$" />
  <orderEntry type="jdk" jdkName="Python 3.9"
  jdkType="Python SDK" />
  <orderEntry type="sourceFolder" forTests="false" />
 </component>
</module>
```

misc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
 <component name="ProjectRootManager"
version="2" project-jdk-name="Python 3.9" project-jdk-
type="Python SDK" />
 <component name="PyCharmProfessionalAdvertiser">
  <option name="shown" value="true" />
 </component>
</project>
```

modules.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
 <component name="ProjectModuleManager">
   <modules>
     <module
fileurl=file://$PROJECT_DIR$/.idea/IBM.iml
filepath="$PROJECT_DIR$/.idea/IBM.iml" />
   </modules>
 </component>
</project>
```

**GITHUB :** IBM-Project-53256-1661325511