REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED.

TEAM ID:PNT2022TMID42258

INTRODUCTION:-

1.1 PROJECT OVERVIEW:

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.

1.2 PURPOSE:

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 Converion 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. LITERATURE SURVEY:-

2.1 EXISTING PROBLEM:

As far, Differently-abled people face discrimination in everyday life. It is difficult for them to communicate with normal people and vice- versa . so many are trying to solve this and give solution by converting sign language into audio and audio(speech) to sign language using Artificial intelligence -neural network,speech recognition,CNN,NLP but it is very tediuos to develop into REAL-TIME APPLICATION.we have tried to provide solutions for this existing solution.

2.2 REFERENCES:

Arun Prasath G,Annapurani,Panaiyappank-Design of an integrated learning approach to assist real-time deaf application using voice recognition system.

Prashant G. Ahire, Kshitija B. Tileka, Tejaswini A. Jawake, Pramod B. WaraleTwo Way Communicator between Deaf and Dumb People and Normal People.

A.Ibarguren,I.Maurtua,B.Sierra-Layered architecture for real time sign recognition: Hand gesture and movement.

Ashish Sethi, Hemanth S,Kuldeep Kumar,Bhaskara Rao N,Krishnan R-SignPro-An Application Suite for Deaf and Dumb.

Ms. Rashmi D. Kyatanavar , Prof. P. R. Futane-Video Gesture Classification using Fourier Descriptors and General Fuzzy Min Max Neural Network.
Shweta S. Shinde,

Rajesh M. Autee,Vitthal K. Bhosale-Real-time two-way communication approach for hearing impaired and dumb person based on image processing.

EriglenGani, AldaKika-Albanian Sign Language (AlbSL) Number Recognition from Both Hand's Gestures Acquired by Kinect Sensors.

Surbhi Rathi,Ujwalla Gawande-Development of full duplex intelligent communication system for deaf and dumb people.

A. Gayathri, A. Sasi Kumar-Sign Language Recognition for Deaf and Dumb

People Using Android Environment.

G. Arun, Prasath,K Annapurani-Real-Time Application for Deaf and Dumb: Bidirectional Communication Using Learning Methods.

<u>2.3 PROBLEM STATEMENT DEFINITION:</u>

Problem Statement (PS)	**I am (Customer)**	**I'm tryingto**	**but**	**Because**	**Which makes mefeel**
PS-1	**DEAF-MUTE PEOPLE**	TO SOCIALIZE WITH OTHERS	THEY CANT COMMUNICATE WITHOUT SIGN LANGUAGE	OF THEIR IMPAIRMENT	UNDERRATED AND INFERIOR TO NON-DISABLED PEOPLE.
PS-2	**NON- DISABLED PEOPLE**	COMMUNICATE WITH DEAF- MUTE PEOPLE	BUT IT VERY COMPLEX TO COMMUNICATE WITH THEM	IT IS DIFFICULT TO LEARNSIGN LANGUAGE	NOT ABLE TO MAKE FRIENDLY CIRCLE WITH THEM

3.IDEATION AND PROPOSED SOLUTION:-

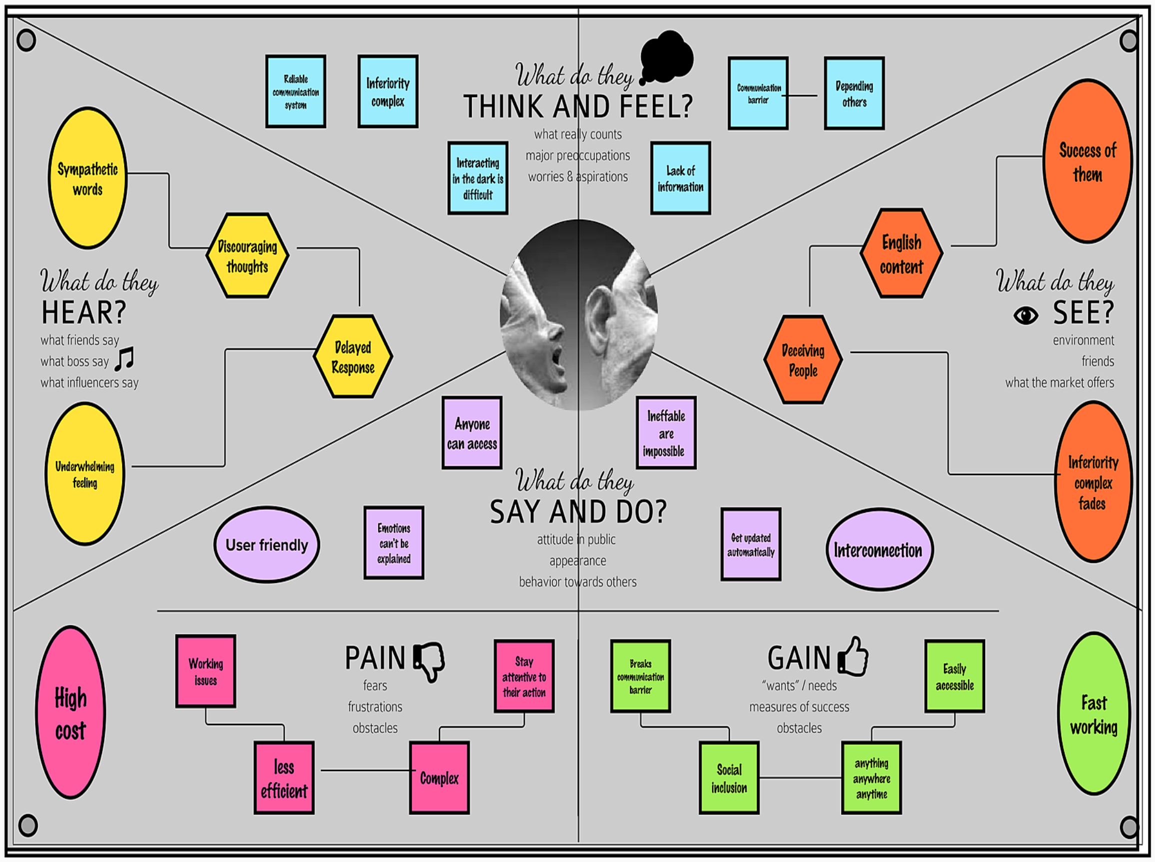
<u>3.1 EMPATHY MAP CANVAS</u>:

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.



Share your feedback

3.2 IDEATION AND BRAINSTORMING:

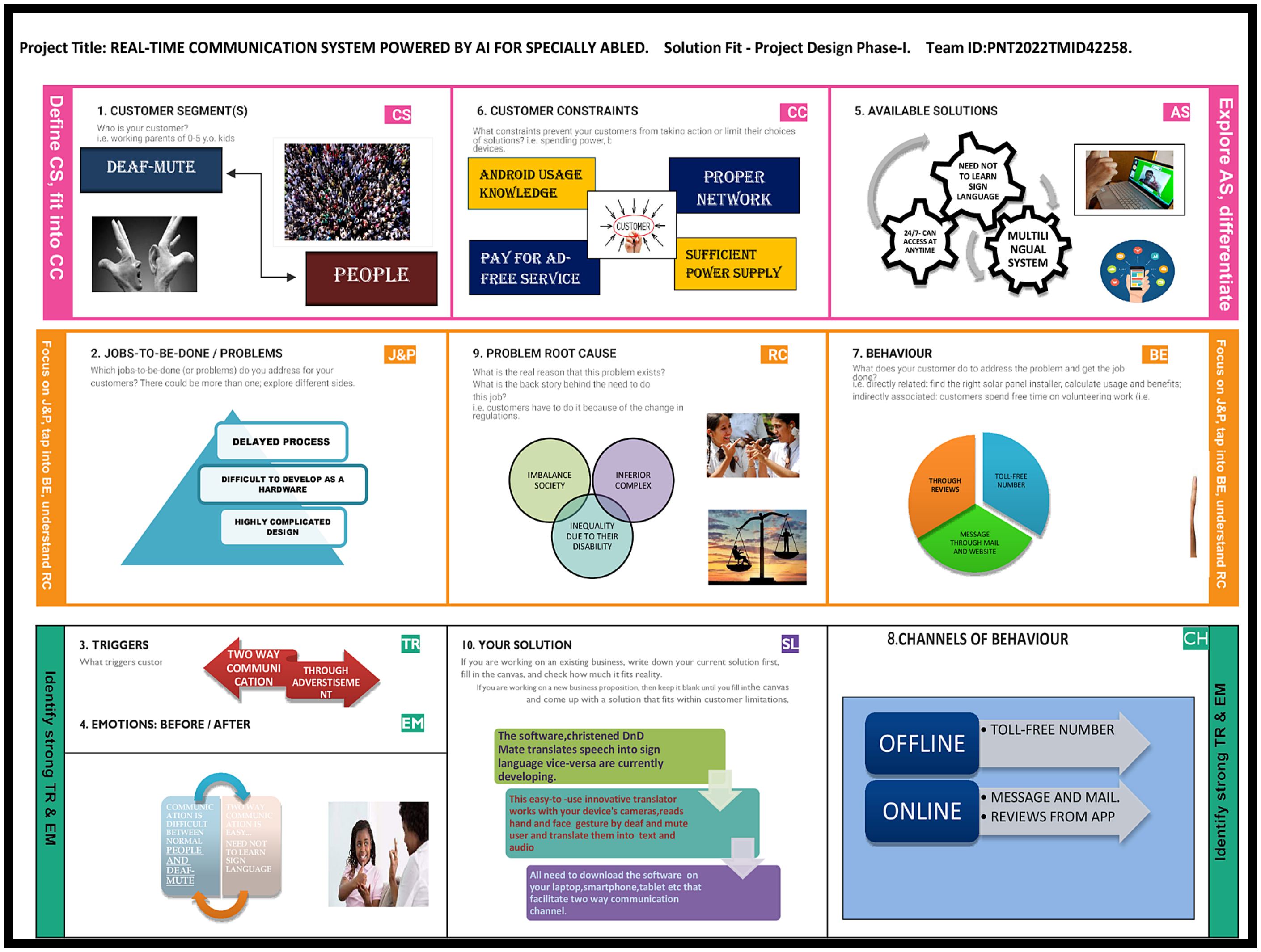
3.3 PROPOSED SOLUTION:

S NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	How might we design and implement the system to overcome the communication barrier between normal people and speech-hearing impaired people?
2.	Idea / Solution description	The project aims to develop a system that converts 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.
3.	Novelty / Uniqueness	1. Enables Two-Way Communication System 2. Accurate Hand Gesture Recognition 3. Retrieval of High-Quality Audio 4. Multilingual system 5. Auto Updation 6. Disabled People get Benefited on a Major Scale 7. Learning Sign Language is not Necessary
4.	Social Impact / Customer Satisfaction	1. Specially Abled People cannot be felt Underrated 2. Easily Accessible by Young and Adults 3. Breaks Inferior Complexity among the Society 4. Portable System 5. Cost Efficient 6. Live Communication 7. Communication Without Delay
5.	Business Model (Revenue	Based on the Popularity of the Application, the number of users gets increased so that Marketing

	Model)	**profit can also be Increased**
6.	**Scalability of the Solution**	**Based on several users who need the service mostly get benefited from this application.This project can be expandable to reach more people which broadens the effectiveness for the users.**

3.4 PROPBLEM SOLUTION FIT:



Project Title: REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED. Solution Fit - Project Design Phase-I. Team ID:PNT2022TMID42258.

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

DEAF-MUTE

PEOPLE

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, b devices.

ANDROID USAGE KNOWLEDGE

PROPER NETWORK

CUSTOMER

PAY FOR AD-FREE SERVICE

SUFFICIENT POWER SUPPLY

5. AVAILABLE SOLUTIONS AS

NEED NOT TO LEARN SIGN LANGUAGE

24/7- CAN ACCESS AT ANYTIME

MULTILINGUAL SYSTEM

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.

DELAYED PROCESS

DIFFICULT TO DEVELOP AS A HARDWARE

HIGHLY COMPLICATED DESIGN

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.

IMBALANCE SOCIETY

INFERIOR COMPLEX

INEQUALITY DUE TO THEIR DISABILITY

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.

THROUGH REVIEWS

TOLL-FREE NUMBER

MESSAGE THROUGH MAIL AND WEBSITE

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

Identify strong TR & EM

3. TRIGGERS TR

What triggers custor

TWO WAY COMMUNICATION

THROUGH ADVERSTISEMENT

4. EMOTIONS: BEFORE / AFTER EM

COMMUNICATION IS DIFFICULT BETWEEN NORMAL PEOPLE AND DEAF-MUTE

TWO WAY COMMUNICATION IS EASY... NEED NOT TO LEARN SIGN LANGUAGE

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 inthe canvas and come up with a solution that fits within customer limitations,

The software,christened DnD Mate translates speech into sign language vice-versa are currently developing.

This easy-to -use innovative translator works with your device's cameras,reads hand and face gesture by deaf and mute user and translate them into text and audio

All need to download the software on your laptop,smartphone,tablet etc that facilitate two way communication channel.

8.CHANNELS OF BEHAVIOUR CH

OFFLINE
- TOLL-FREE NUMBER

ONLINE
- MESSAGE AND MAIL.
- REVIEWS FROM APP

Identify strong TR & EM

4.REQUIREMENT ANALYSIS:-

4.1 FUNCTIONAL REQUIREMENTS:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration**	**Registration through website Registration through Email Id and mobile number Registration through installing the application**
FR-2	**User Confirmation**	**Confirmation via Email verification Confirmation via mobile OTP verification**
FR-3	**User Login**	**Login to your account by entering your Email Id or mobile number and password**
FR-4	**User Interface**	**Interaction between the application and the user is made easy**
FR-5	**Account management**	**Easy recovery process by OTP verification by Email or SMS in case the account password is lost**
FR-6	**Database management**	**Storing frequently used sign language for easy communication**

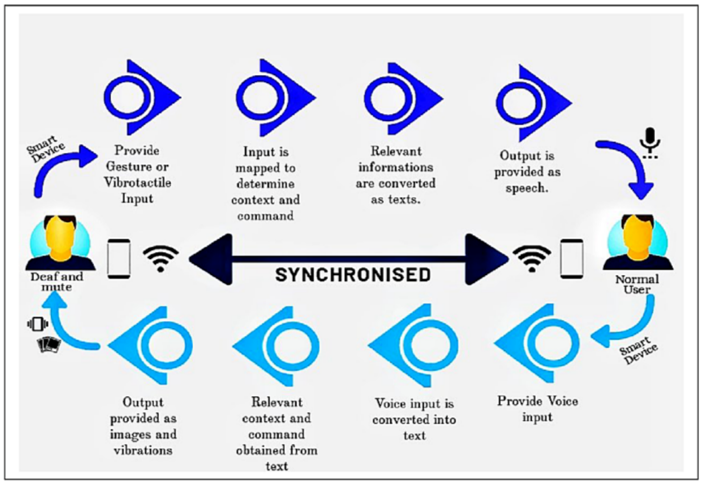
4.2 NON- FUNCTIONAL REQUIREMENTS:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	**Easy to use because of the user-friendly nature of the application**
NFR-2	**Security**	**The account cannot be misused due to the user verification process**
NFR-3	**Reliability**	**Highly reliable due to high accuracy**
NFR-4	**Performance**	**Because of very minimal errors, the performance is very high**

NFR-5	**Availability**	**The application is available to all and can be accessed by every deaf-mute people**
NFR-6	**Scalability**	**The application can be easily modified based on the user's preferences**

5.PROJECT DESIGN:-

5.1 DATA FLOW DIAGRAMS:



Smart Device
Provide Gesture or Vibrotactile Input
Input is mapped to determine context and command
Relevant informations are converted as texts.
Output is provided as speech.
Deaf and mute
SYNCHRONISED
Normal User
Smart Device
Output provided as images and vibrations
Relevant context and command obtained from text
Voice input is converted into text
Provide Voice input

5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Normal people and Deaf-mute people	**Registration**	**USN-1**	**As a user, I can register for the application by entering my email, and password, and**	**I can access my account/dashboard**	**High**	**Sprint-1**

			confirming my password			
		USN-2	**As a user, I will receive a confirmation email once I have registered for the application**			**Sprint-1**
	Dashboard		**Two options available** **Choose the option based on who uses the app.** **If you are normal person click the "normal people" option.** **If you are deaf-mute people click "deaf-mute" Option**		**High**	**Sprint-1**
Normal people			**Give access to camera to recognize the gestures** **Give access to**		**High**	**Sprint-1**

			microphone to give our message through voice			
Deaf-mute people			Give access to display to view the message sent by normal people.		High	Sprint-1

5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

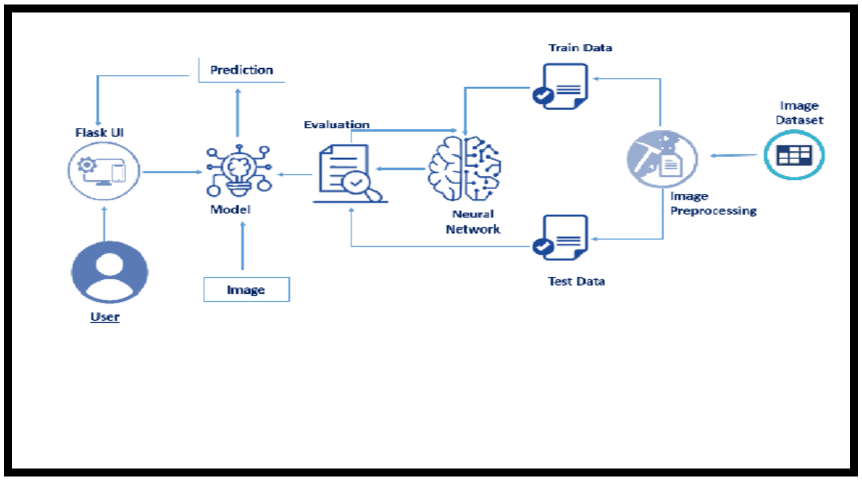


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1		User can interact using	HTML

	User Interface	Web UI and Mobile Application	
2	Application Logic-1	Get the image dataset and pre-process the images which will be used for building the model	Python
3	Application Logic-2	Building and testing the model	IBM Watson STT service
4	Application Logic-3	Building the application	Flask
5	Database	Data Type, Configurations etc	MySQL, NoSQL, etc.
6	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
9	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S. No	**Component**	**Description**	**Technology**
1	Security Implementations	use of firewalls	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc
2	Scalable Architecture	It can handle business growth	Web server tuning,Operating system

			tuning
3	Availability	Available to everyone	Open source framework
4	Performance	Without delay outputs can be viewed	Deep Neural network

6.PROJECT PLANNING AND SCHEDULING:-

6.1 SPRINT PLANNING AND ESTIMATION:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature Survey & Literature survey on the 10 SEPTEMBER 2022 Information Gathering selected project & gathering information by referring the, technical papers, research publications etc	18 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas 10 SEPTEMBER 2022 to capture the user Pains & Gains, Prepare list of problem statements	18 SEPTEMBER 2022
Ideation	List the by organizing the 17 SEPTEMBER 2022 brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	23 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability	24 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit Problem Solution Fit do	05 OCTOBER 2022
Solution Architecture	Prepare solution architecture Solution Architecture document	06 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	07 OCTOBER 2022
Functional Requirement	Prepare the functional Functional Requirement document.	10 OCTOBER

		2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	11 OCTOBER 2022
Technology architecture	Prepare the Technology Architecture technology architecture diagram.	1 NOVEMBER 2022
Prepare milestone and activity list	Prepare the milestones & Prepare Milestone & Activity activity list	15 NOVEMBER 2022
Project Development - Delivery of Sprint1, 2,3,4	Sprint 1-Collecting the data and processing the image Sprint 2-Testing the model Sprint 3-Building the application Sprint 4-Training the CNN model on IBM cloud	15 NOVEMBER 2022 15 NOVEMBER 2022 15 NOVEMBER 2022 In Progress...

6.2 SPRINT DELIVERY SCHEDULE:

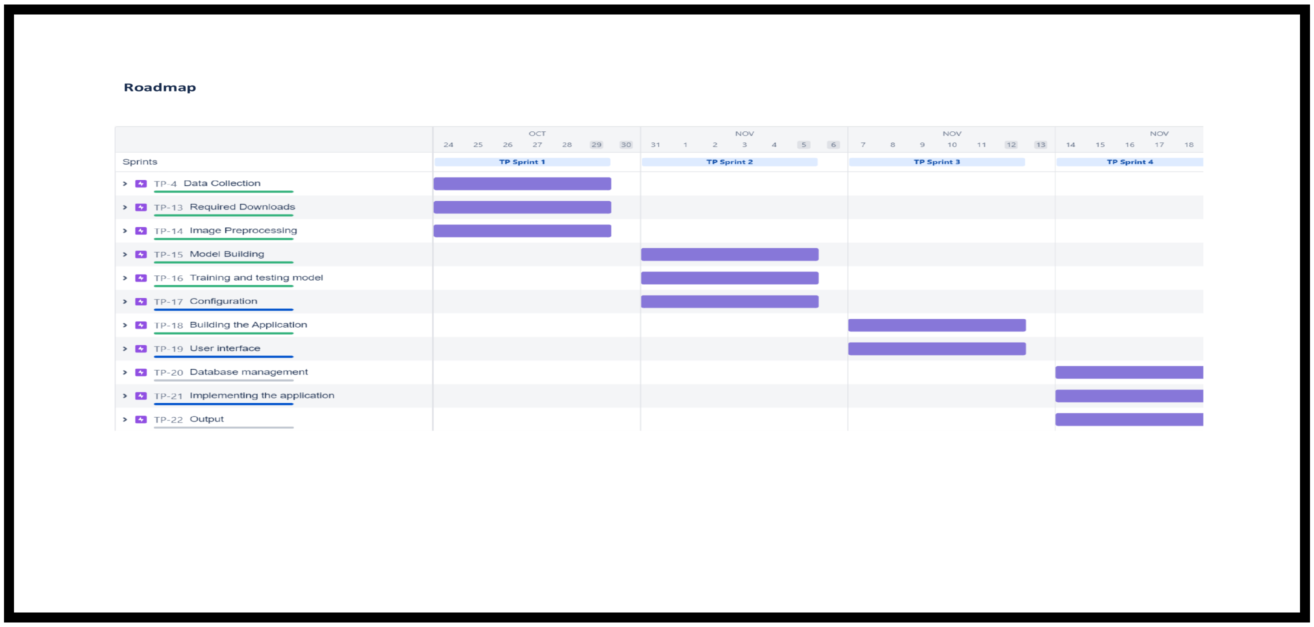
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collecting the required dataset for the project.	8	High	Kishore
Sprint-1	Required Downloads	USN-2	Downloading all the tools that are required for the project.	3	Low	Shuruthi
Sprint-2	Model Building	USN-3	Building the project model by downloading	2	High	Anandha

			required Python libraries.			
Sprint-1	Image Preprocessing	USN-1	Image data generation and training and testing the dataset.	8	Medium	Rebecca
Sprint-2	Training and testing the model	USN-2	Training the created project model and testing it.	5	Medium	Kishore
Sprint-2	Configuration	USN-1	Two options available- Choose the option based on who uses the app. If you are a normal person click the "normal people" option.	13	High	Anandha
Sprint -3	Building the Application	USN-2	Building the final application using flask and HTML.	8	High	Rebecca
Sprint-3	User interface	USN-1	Give access to the microphone to give our message through voice	5	Low	Kishore
Sprint-4		USN-3	Give access to	3		Shuruthi

	Database management		the display to view the message sent by normal people		Medium	
Sprint - 4	Output	USN-3	Audio gets converted into sign language	8	High	Shuruthi

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	19	6 Days	24 Oct 2022	29 Oct 2022	19	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	13	6 Days	07 Nov 2022	12 Nov 2022	13	12 Nov 2022
Sprint-4	19	6 Days	14 Nov 2022	19 Nov 2022	19	19 Nov 2022

6.3 REPORTS FROM JIRA:



7.CODING AND SOLUTIONING:-

7.1 FEATURE 1:

```
pip install keras==2.10.0

Collecting keras==2.10.0
  Using cached keras-2.10.0-py2.py3-none-any.whl (1.7 MB)
Installing collected packages: keras
  Attempting uninstall: keras
    Found existing installation: keras 2.7.0
    Uninstalling keras-2.7.0:
      Successfully uninstalled keras-2.7.0
Successfully installed keras-2.10.0
Note: you may need to restart the kernel to use updated packages.
```

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2,
zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)

import tensorflow as tf
def get_data(arg1, **kwargs):
  tf.keras.preprocessing.image_dataset_from_directory
 labels="inferred",
    label_mode="int",
    class_names=None,
    color_mode="rgb",
    batch_size=32,
    image_size=(256, 256),
    shuffle=True,
    seed=None,
    validation_split=None,
    subset=None,
    interpolation="bilinear",
    follow_links=False,
    crop_to_aspect_ratio=False,
    **kwargs
)
def get_data(path):
  tf.keras.preprocessing.image.load_img(
    path, grayscale=False, color_mode="rgb", target_size=None,
interpolation="nearest"
)
def get_data(image_path):
  image = tf.keras.preprocessing.image.load_img(image_path)
  input_arr = tf.keras.preprocessing.image.img_to_array(image)
  input_arr = np.array([input_arr])  # Convert single image to a batch.
  predictions = model.predict(input_arr)
def get_data(img):
  tf.keras.preprocessing.image.img_to_array(img, data_format=None,
dtype=None)
from tensorflow.python.keras.utils.np_utils import to_categorical
import numpy as np
from PIL import Image
img_data = np.random.random(size=(100, 100, 3))
img = tf.keras.preprocessing.image.array_to_img(img_data)
array = tf.keras.preprocessing.image.img_to_array(img)
x_train=train_datagen.flow_from_directory
```

```
Found 15750 images belonging to 9 classesx_test =
test_datagen.flow_from_directory(r'C:\Users\pragadeswar\Downloads\conversat
ion engine for deaf and dumb\Dataset\test_set', target_size = (64, 64),
batch_size = 300, class_mode = 'categorical', color_mode = 'grayscale')
Found 2250 images belonging to 9 classes.from keras.models import
Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
model = Sequential()
model.add(Convolution2D(32, (3, 3), input_shape = (64, 64, 1), activation
= 'relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Flatten())
model.add(Dense(units = 512, activation = 'relu'))
model.add(Dense(units = 9, activation = 'softmax'))
model.compile(loss = 'categorical_crossentropy', optimizer = 'adam',
metrics = ['accuracy'])
model.fit_generator(x_train, steps_per_epoch = 24, epochs = 10,
validation_data = x_test, validation_steps = 40
model.save('aslpng1.h5')
from keras.models import load_model
import numpy as np
import cv2
model = load_model('aslpng1.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
  prediction = model.predict(img)
  print(prediction)
  predictions = (model.predict(img) > 0.5).astype("int32")
  print(prediction)
import sys
sys.setrecursionlimit(1500)
frame = cv2.imread
data = detect(frame)
```

7.2 FEATURE 2:

UPLOAD HTML CODE:

```
<html lang="en">
<head>
  <title>Conversation Engine</title>
  <link
href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
<style>
.header {
position: relative;
top:0;
margin:0px;
z-index: 1;
left: 0px;
right: 0px;
position: fixed;
background-color: #F36262 ;
color: white;
box-shadow: 0px 8px 4px grey;
overflow: hidden;
padding-left:20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:8%;
text-align: center;
}
.topnav {
 overflow: hidden;
 background-color: #FCAD98;
}
.topnav-right a {
 float: left;
 color: black;
 text-align: center;
 padding: 14px 16px;
 text-decoration: none;
 font-size: 18px;
}
.topnav-right a:hover {
 background-color: #FCAD98;
 color: black;
}
.topnav-right a.active {
 background-color: #FCAD98;
 color: white;
}
.topnav-right {
 float: right;
 padding-right:100px;
}
body {
 background-color: ;
 background-repeat: no-repeat;
 background-size:cover;
 background-image:
url("https://i.pinimg.com/originals/b2/1d/c6/b21dc69346915015bc4e19bd502f
401b.gif");
  background-size: cover;
 background-position: 0px 0px;
 }
 .button {
 background-color: #091425;
 border: none;
 color: white;
```

```
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 12px;
  border-radius: 16px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  display: inline-block;
  margin-bottom:18px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}
button {
  background-color: #091425;
  color: white;
  padding: 14px 20px;
  margin-bottom:10px;
  border: none;
  cursor: pointer;
  width: 17%;
  border-radius:4px;
  font-family:Montserrat;
}
button:hover {
  opacity: 0.8;
}
.cancelbtn {
  width: auto;
  padding: 10px 18px;
```

```
 background-color: #f44336;
}
.imgcontainer {
 text-align: center;
 margin: 24px 0 12px 0;
}
img.avatar {
 width: 30%;
 border-radius: 50%;
}
.container {
 padding: 16px;
}
span.psw {
 float: right;
 padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
 span.psw {
   display: block;
   float: none;
 }
 .cancelbtn {
   width: 100%;
 }
}
.home{
margin:80px;
width: 84%;
 height: 500px;
 padding-top:10px;
 padding-left: 30px;
}
.login{
margin:80px;
```

```
box-sizing: content-box;
  width: 84%;
  height: 420px;
  padding: 30px;
  border: 10px solid blue;
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
 border: 10px solid blue;
}
.mySlides {display: none;}
img {vertical-align: middle;}
/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}
/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
 display: inline-block;
 transition: background-color 0.6s ease;
}
.active {
 background-color: #FCAD98;
}
/* Fading animation */
.fade {
 -webkit-animation-name: fade;
 -webkit-animation-duration: 1.5s;
 animation-name: fade;
 animation-duration: 1.5s;
}
@-webkit-keyframes fade {
 from {opacity: .4}
 to {opacity: 1}
}
@keyframes fade {
 from {opacity: .4}
 to {opacity: 1}
}
/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
 .text {font-size: 11px}
}
.bar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
```

```
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
p
{
color:black;
font-style:italic;
font-size:30px;
}
</style>
</head>
<body style="background-
image:url({{url_for('static',filename='images/bck3.png')}});background-
position: center;background-repeat: no-repeat;
background-size: cover;">
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black;
padding-top:1%;padding-left:5%;">Real Time Communication System for
Deaf & Dumb</div>
 <div class="topnav-right"style="padding-top:0.5%;">
  <a  href="/home">Home</a>
  <a class="active" href="/upload">Open Web Cam</a>
 </div>
</div>
</body>
```

BACKEND:

```
# USAGE
# import the necessary packages
from flask import Flask,render_template,request
# Flask-It is our framework which we are going to use to run/serve our
application.
#request-for accessing file which was uploaded by the user on our
application.
import cv2 # opencv library
from tensorflow.python.keras.models import load_model#to load our
trained model
import numpy as np
from gtts import gTTS #to convert text to speech
from skimage.transform import resize
import os
from keras.preprocessing import image
from playsound import playsound
'''
def playaudio(text):
  speech=gTTS(text)
  print(type(speech))
  speech.save("output1.mp3")
  playsound("output1.mp3")
  return
'''
app = Flask(_name_,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model('aslpng1.h5')
print("Loaded model from disk")
vals = ['A', 'B','C','D','E','F','G','H','I']
#app=Flask(_name_,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
  return render_template('home.html')
@app.route('/home', methods=['GET'])
```

```
def home():
   return render_template('home.html')
@app.route('/upload', methods=['GET', 'POST'])
def predict():
      # Get a reference to webcam #0 (the default one)
      print("[INFO] starting video stream...")
      vs = cv2.VideoCapture(0)
      #writer = None
      (W, H) = (None, None)
 # loop over frames from the video file stream
      while True:
# read the next frame from the file
        (grabbed, frame) = vs.read()
# if the frame was not grabbed, then we have reached the end
# of the stream
        if not grabbed:
          break
# if the frame dimensions are empty, grab them
        if W is None or H is None:
          (H, W) = frame.shape[:2]
# clone the output frame, then convert it from BGR to RGB
#ordering and resize the frame to a fixed 64x64
        output = frame.copy()
        #print("apple")
        img = resize(frame,(64,64,1))
        img = np.expand_dims(img,axis=0)
        if(np.max(img)>1):
          img = img/255.0
        result = np.argmax(model.predict(img), axis=-1)
        index=['A', 'B','C','D','E','F','G','H','I']
        result=str(index[result[0]])
        #print(result)
        #result=result.tolist()
        cv2.putText(output, "It indicates: {}".format(result), (10, 120),
cv2.FONT_HERSHEY_PLAIN,
              2, (0,255,255), 1)
        #converts text to speech and plays the audio
        speech = gTTS(text = result, lang = 'en', slow = False)
        #speech=gTTS(text)
        print(type(speech))
        speech.save("text.mp3")
        os.system("start text.mp3")
        cv2.imshow("Output", output)
        key = cv2.waitKey(1) & 0xFF
# if the `q` key was pressed, break from the loop
        if key == ord("q"):
          break
      # release the file pointers
      print("[INFO] cleaning up...")
      vs.release()
      cv2.destroyAllWindows()
      return render_template("upload.html")
if _name_ == '_main_':
   app.run(host='0.0.0.0', port=8000, debug=False)
```

HOME HTML :

```
<html>
<script>
</script>
<style>
.header {
position: relative;
top:0;
margin:0px;
z-index: 1;
left: 0px;
right: 0px;
position: fixed;
background-color: #FCAD98 ;
color: white;
box-shadow: 0px 8px 2px grey;
overflow: hidden;
padding-left:20px;
```

```
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:8%;
text-align: center;
}
.topnav {
 overflow: hidden;
 background-color: #FCAD98;
}
.topnav-right a {
 float: left;
 color: black;
 text-align: center;
 padding: 14px 16px;
 text-decoration: none;
 font-size: 18px;
}
.topnav-right a:hover {
 background-color: #FCAD98;
 color: black;
}
.topnav-right a.active {
 background-color: #FCAD98;
 color: white;
}
.topnav-right {
 float: right;
 padding-right:100px;
}
body {
background-image: -webkit-linear-gradient(90deg, skyblue 0%, steelblue
100%);
 background-image: url("");
  background-size: cover;
 background-attachment: fixed;
 background-size: 100% 100%;
 background-color: ;
 background-repeat: no-repeat;
 background-size:cover;
 background-position: 0px 0px;
 }
 .button {
 background-color: #091425;
 border: none;
 color: white;
 padding: 15px 32px;
 text-align: center;
 text-decoration: none;
 display: inline-block;
 font-size: 12px;
border-radius: 16px;
}
.button:hover {
 box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}
input[type=text], input[type=password] {
 width: 100%;
 padding: 12px 20px;
 display: inline-block;
 margin-bottom:18px;
 border: 1px solid #ccc;
 box-sizing: border-box;
}
button {
 background-color: #091425;
 color: white;
 padding: 14px 20px;
 margin-bottom:10px;
 border: none;
 cursor: pointer;
 width: 17%;
 border-radius:4px;
 font-family:Montserrat;
}
button:hover {
 opacity: 0.8;
}
.cancelbtn {
 width: auto;
 padding: 10px 18px;
 background-color: #f44336;
}
.imgcontainer {
 text-align: center;
 margin: 24px 0 12px 0;
}
img.avatar {
 width: 30%;
 border-radius: 50%;
}
.container {
 padding: 16px;
}
span.psw {
 float: right;
 padding-top: 16px;
}
/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
 span.psw {
   display: block;
   float: none;
 }
 .cancelbtn {
   width: 100%;
 }
}
.home{
margin:80px;
```

```
  width: 84%;
  height: 500px;
  padding-top:10px;
  padding-left: 30px;
}
.login{
margin:80px;
box-sizing: content-box;
  width: 84%;
  height: 420px;
  padding: 30px;
  border: 10px solid blue;
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
 border: 10px solid blue;
}
.mySlides {display: none;}
img {vertical-align: middle;}
/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}
/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
```

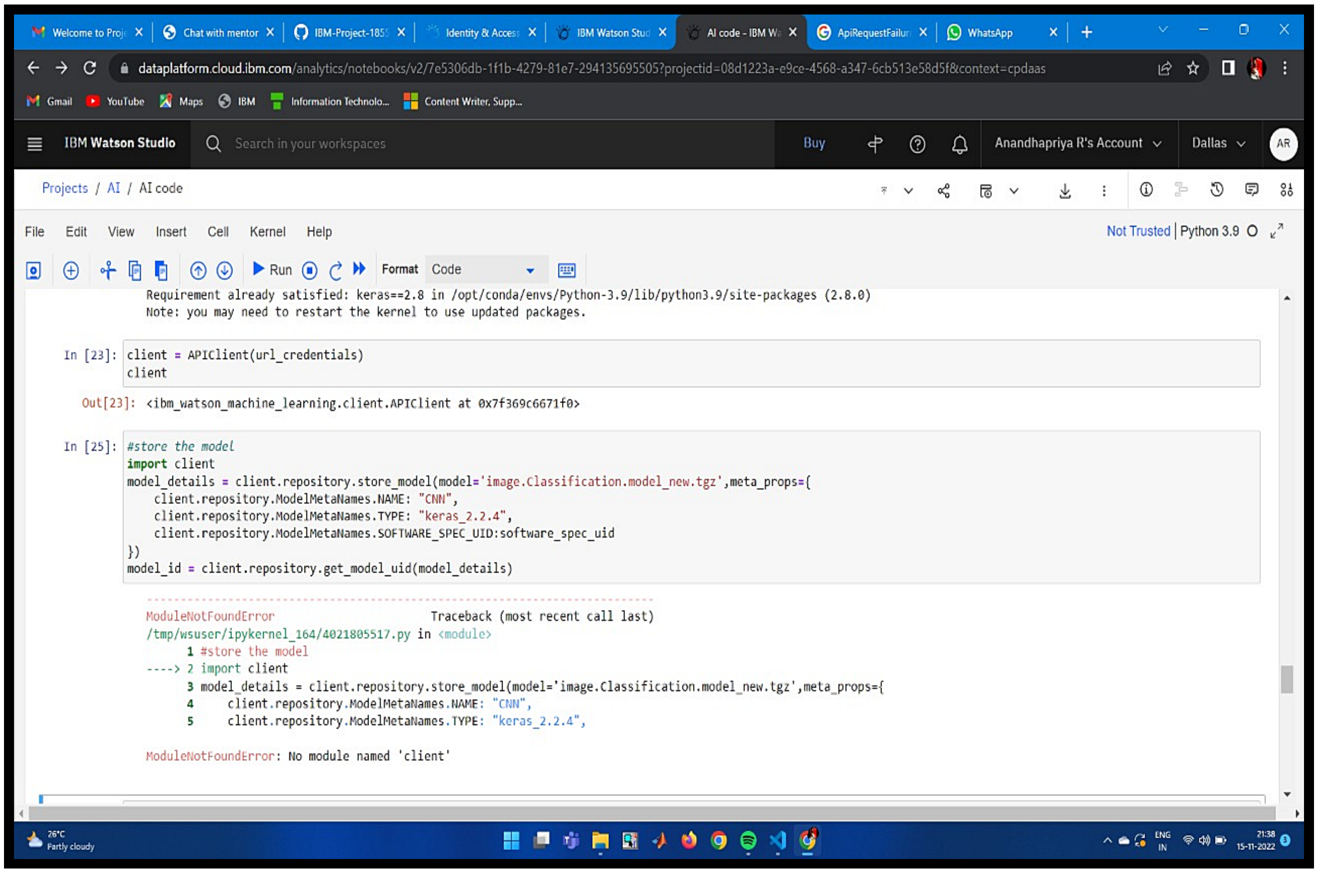
```
/* The dots/bullets/indicators */
.dot {
 height: 15px;
 width: 15px;
 margin: 0 2px;
 background-color: #bbb;
 border-radius: 50%;
 display: inline-block;
 transition: background-color 0.6s ease;
}
.active {
 background-color: #FCAD98;
}
/* Fading animation */
.fade {
 -webkit-animation-name: fade;
 -webkit-animation-duration: 1.5s;
 animation-name: fade;
 animation-duration: 1.5s;
}
@-webkit-keyframes fade {
 from {opacity: .4}
 to {opacity: 1}
}
@keyframes fade {
 from {opacity: .4}
 to {opacity: 1}
}
/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
 .text {font-size: 11px}
}
@import
url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');
* {
 box-sizing: border-box;
}
```

```
body {
 min-height: 100vh;
 margin: 0;
 color: #fff;
 font-family: 'Poppins',sans-serif;
 display: flex;
 align-items: center;
 justify-content: center;
 background-color: #f5f5f5;
}
.container {
 max-width: 1376px;
 margin: auto;
 padding: 2rem 1.5rem;
}
.cards {
 display: flex;
 flex-wrap: wrap;
 align-items: center;
 justify-content: center;
}
.card {
 cursor: pointer;
 background-color: transparent;
 height: 300px;
 perspective: 1000px;
 margin: 1rem;
 align-items: center;
 justify-content: center;
}
.card h3 {
 border-bottom: 1px #fff solid;
 padding-bottom: 10px;
 margin-bottom: 10px;
 text-align: center;
 font-size: 1.6rem;
 word-spacing: 3px;
```

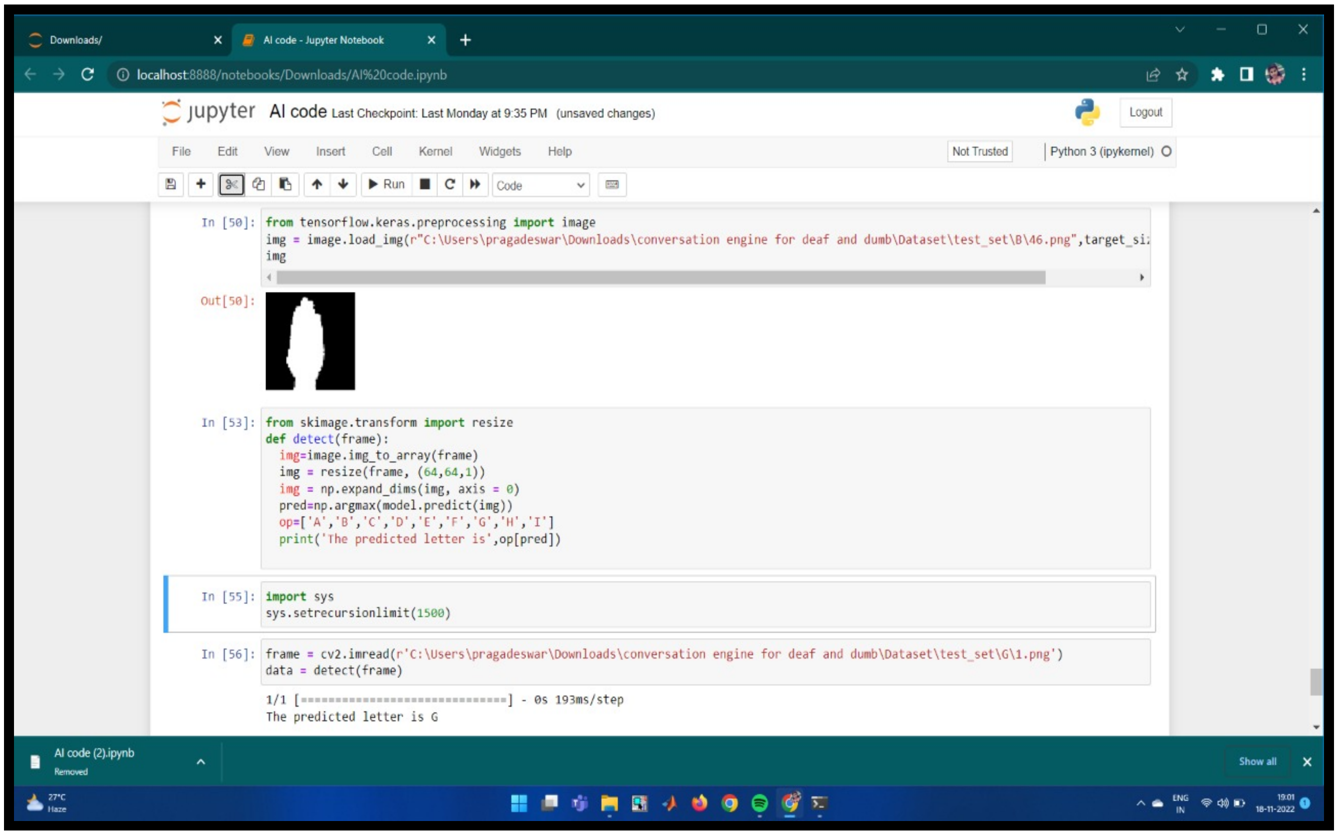
```
}
.card p{
 opacity: 0.75;
 font-size: 0.8rem;
 line-height: 1.4;
}
.card img {
 width: 360px;
 height: 300px;
 object-fit: cover;
 border-radius: 3px;
}
.card-inner {
 position: relative;
 width: 360px;
 height: 100%;
 transition: transform 0.9s;
 transform-style: preserve-3d;
}
.card:hover .card-inner {
 transform: rotateY(180deg);
}
.card-front,
.card-back {
 position: absolute;
 width: 360px;
 height: 100%;
 -webkit-backface-visibility: hidden;
 backface-visibility: hidden;
}
.card-back {
 background-color: #222;
 color: #fff;
 padding: 1.5rem;
 transform: rotateY(180deg);
}
.text-block {
 position: absolute;
 bottom: 20px;
 right: 20px;
 background-color: black;
 color: white;
 padding-left: 20px;
 padding-right: 20px;
}
p
{
color:black;
font-style:italic;
font-size:30px;
}
</style>
<body style="background-
image:url({{url_for('static',filename='images/bck3.png')}});background-
position: center;background-repeat: no-repeat;
background-size: cover;">
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black;
padding-top:1%;padding-left:5%;">Real Time Communication System for
Deaf & Dumb</div>
 <div class="topnav-right"style="padding-top:0.5%;">
  <a class="active" href="/home">Home</a>
  <a href="/upload">Open Web Cam</a>
 </div>
</div>
<div class="container">
  <p>In our society, we have people with disabilities. The technology is
developing day by day but no significant developments are undertaken for
the betterment of these people. Communication 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.</p>
</div>
</body>
<html>

<u>OUTPUT:</u>

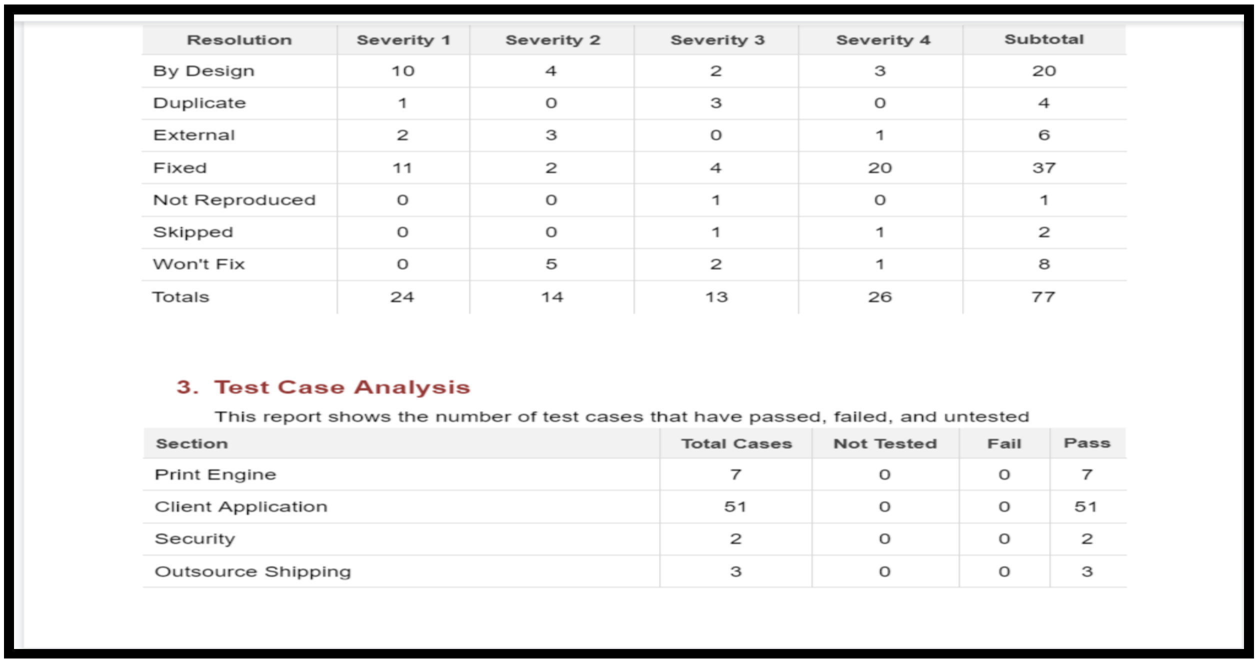


```
    Requirement already satisfied: keras==2.8 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.8.0)
    Note: you may need to restart the kernel to use updated packages.

In [23]: client = APIClient(url_credentials)
         client

Out[23]: <ibm_watson_machine_learning.client.APIClient at 0x7f369c6671f0>

In [25]: #store the model
         import client
         model_details = client.repository.store_model(model='image.Classification.model_new.tgz',meta_props={
             client.repository.ModelMetaNames.NAME: "CNN",
             client.repository.ModelMetaNames.TYPE: "keras_2.2.4",
             client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid
         })
         model_id = client.repository.get_model_uid(model_details)

    ---------------------------------------------------------------------------
    ModuleNotFoundError                       Traceback (most recent call last)
    /tmp/wsuser/ipykernel_164/4021805517.py in <module>
          1 #store the model
    ----> 2 import client
          3 model_details = client.repository.store_model(model='image.Classification.model_new.tgz',meta_props={
          4     client.repository.ModelMetaNames.NAME: "CNN",
          5     client.repository.ModelMetaNames.TYPE: "keras_2.2.4",

    ModuleNotFoundError: No module named 'client'
```



```
In [50]: from tensorflow.keras.preprocessing import image
         img = image.load_img(r"C:\Users\pragadeswar\Downloads\conversation engine for deaf and dumb\Dataset\test_set\B\46.png",target_si
         img

Out[50]:

In [53]: from skimage.transform import resize
         def detect(frame):
           img=image.img_to_array(frame)
           img = resize(frame, (64,64,1))
           img = np.expand_dims(img, axis = 0)
           pred=np.argmax(model.predict(img))
           op=['A','B','C','D','E','F','G','H','I']
           print('The predicted letter is',op[pred])

In [55]: import sys
         sys.setrecursionlimit(1500)

In [56]: frame = cv2.imread(r'C:\Users\pragadeswar\Downloads\conversation engine for deaf and dumb\Dataset\test_set\G\1.png')
         data = detect(frame)

         1/1 [==============================] - 0s 193ms/step
         The predicted letter is G
```

8. TESTING:-

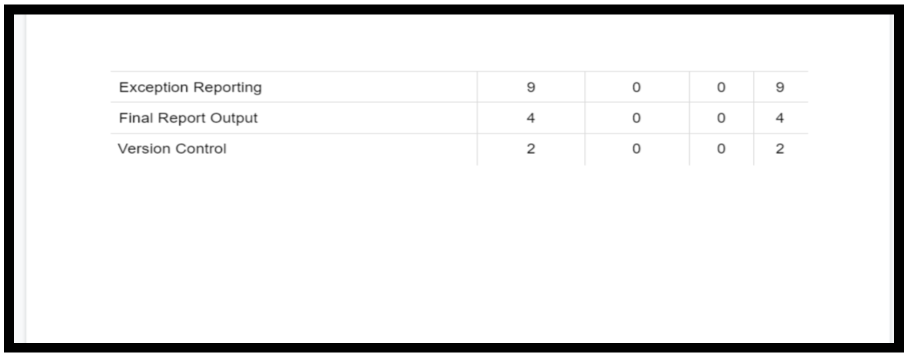


Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3. Test Case Analysis

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

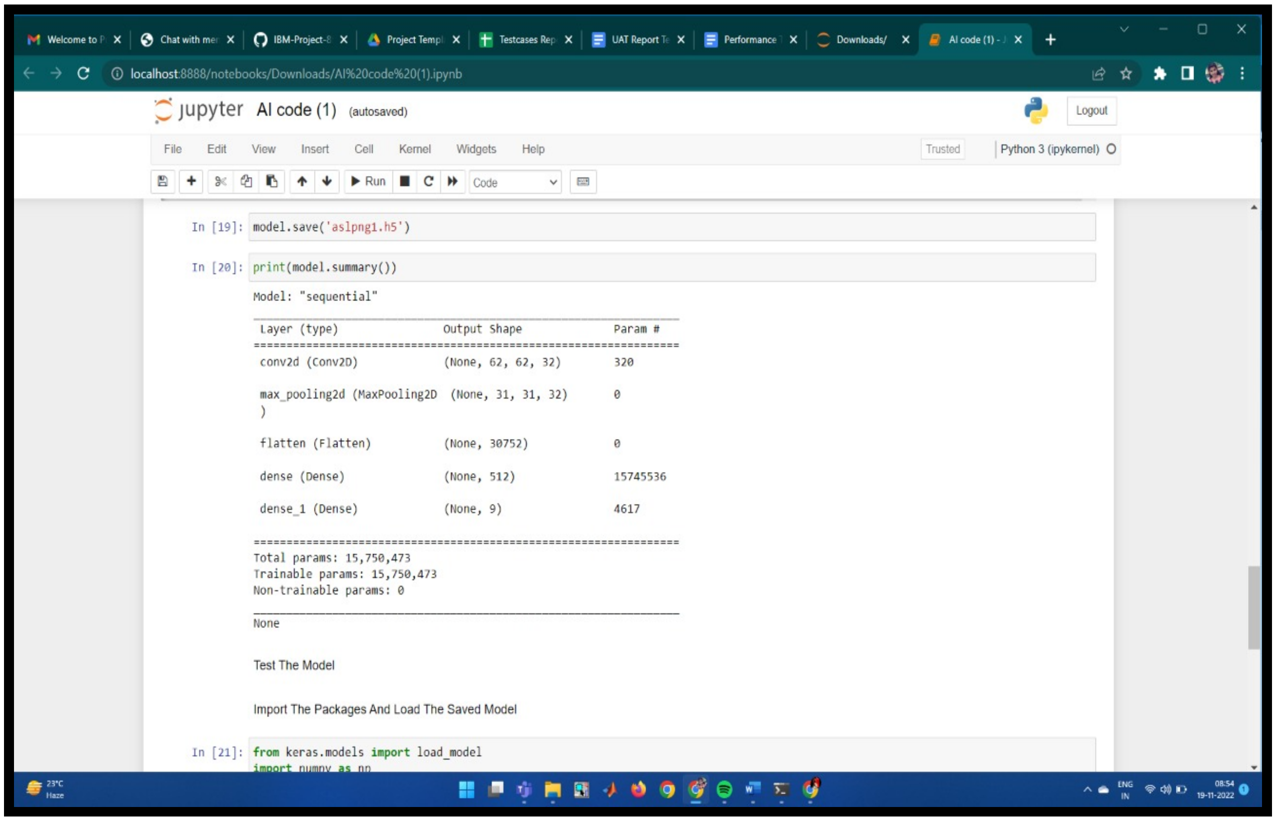
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3



Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.RESULTS:-

9.1 PERFORMANCE METRICS:

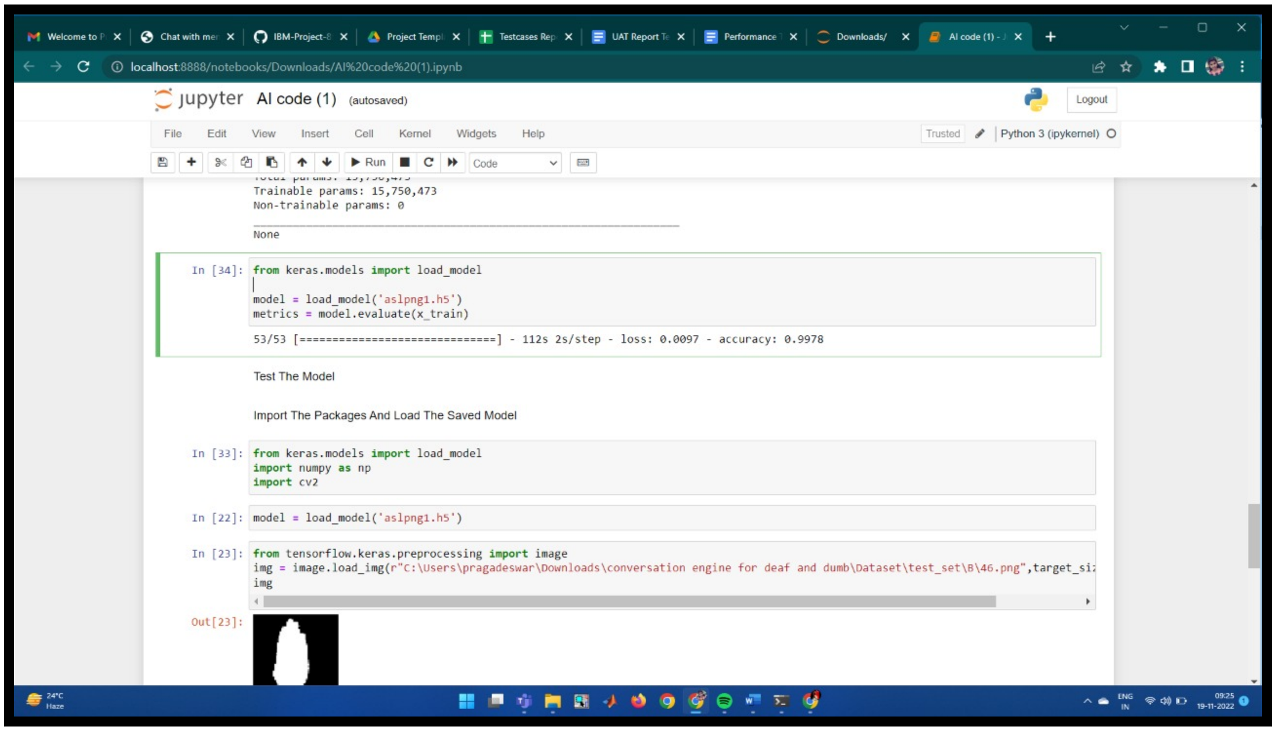

localhost:8888/notebooks/Downloads/AI%20code%20(1).ipynb
jupyter AI code (1) (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [19]: model.save('aslpng1.h5')

In [20]: print(model.summary())

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        320

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 flatten (Flatten)           (None, 30752)             0

 dense (Dense)               (None, 512)               15745536

 dense_1 (Dense)             (None, 9)                 4617

=================================================================
Total params: 15,750,473
Trainable params: 15,750,473
Non-trainable params: 0
_________________________________________________________________
None
```

Test The Model

Import The Packages And Load The Saved Model

```
In [21]: from keras.models import load_model
         import numpy as np
```



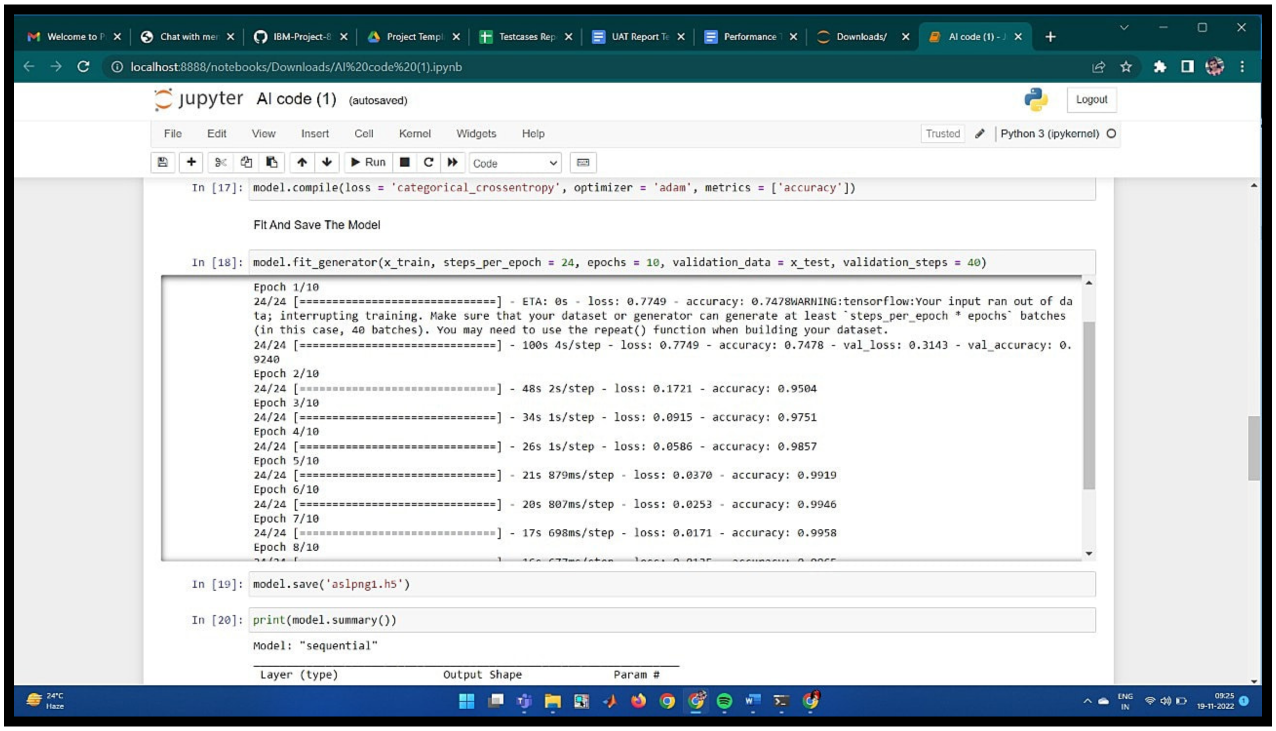
```
Trainable params: 15,750,473
Non-trainable params: 0
_________________________________________________________________
None
```

```
In [34]: from keras.models import load_model

         model = load_model('aslpng1.h5')
         metrics = model.evaluate(x_train)

         53/53 [==============================] - 112s 2s/step - loss: 0.0097 - accuracy: 0.9978
```

Test The Model

Import The Packages And Load The Saved Model

```
In [33]: from keras.models import load_model
         import numpy as np
         import cv2

In [22]: model = load_model('aslpng1.h5')

In [23]: from tensorflow.keras.preprocessing import image
         img = image.load_img(r"C:\Users\pragadeswar\Downloads\conversation engine for deaf and dumb\Dataset\test_set\B\46.png",target_si
         img

Out[23]:
```



```
In [17]: model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
```

Fit And Save The Model

```
In [18]: model.fit_generator(x_train, steps_per_epoch = 24, epochs = 10, validation_data = x_test, validation_steps = 40)

         Epoch 1/10
         24/24 [==============================] - ETA: 0s - loss: 0.7749 - accuracy: 0.7478WARNING:tensorflow:Your input ran out of da
         ta; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` batches
         (in this case, 40 batches). You may need to use the repeat() function when building your dataset.
         24/24 [==============================] - 100s 4s/step - loss: 0.7749 - accuracy: 0.7478 - val_loss: 0.3143 - val_accuracy: 0.
         9240
         Epoch 2/10
         24/24 [==============================] - 48s 2s/step - loss: 0.1721 - accuracy: 0.9504
         Epoch 3/10
         24/24 [==============================] - 34s 1s/step - loss: 0.0915 - accuracy: 0.9751
         Epoch 4/10
         24/24 [==============================] - 26s 1s/step - loss: 0.0586 - accuracy: 0.9857
         Epoch 5/10
         24/24 [==============================] - 21s 879ms/step - loss: 0.0370 - accuracy: 0.9919
         Epoch 6/10
         24/24 [==============================] - 20s 807ms/step - loss: 0.0253 - accuracy: 0.9946
         Epoch 7/10
         24/24 [==============================] - 17s 698ms/step - loss: 0.0171 - accuracy: 0.9958
         Epoch 8/10

In [19]: model.save('aslpng1.h5')

In [20]: print(model.summary())

         Model: "sequential"
         _________________________________________________________________
          Layer (type)                Output Shape              Param #
```

10.ADVANTAGES AND DISADVANTAGES:-

PROS:

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. Can access the app all time whenever we need to communicate. portable anywhere and also less maintenance cost.

CONS:

Complex to make a live communication because some delay occurs to make a two way communication. Works only in ONLINE mode.so network facility is needed all time.Tedious to setup sign language information. Requires battery power .

11.CONCLUSION:-

The project aims to develop an AI model that converts sign language into speech that can be understandable by people and vice-versa to specially abled people.This model brings changes among our society and build commmunication between specially challenged people and normal people and breaks the discrimination shown towards specially abled people and they will not feel underrated in our society.

12.FUTURE SCOPE:-

Real-time communication has been impacting global industries to procure 100 times productivity and revenue. Information exchange is quicker and more efficient, and the emergence of push notifications can even accelerate the entire communication process.

Messaging platform can simplify your operations by connecting your internal and external departments and consolidating all data and conversation history into channels. Consequently, this gives you and your team uninhibited access to information and discussions necessary to make well-informed choices.Hence people can be highly benefited through this process

13.APPENDIX:-

GitHub LINK : https://github.com/IBM-EPBL/IBM-Project-858-1658326949

DEMO LINK: https://youtu.be/YQLemdYH4YU