REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

A PROJECT REPORT

Submitted by

ANANDH P	422419104005
ARUN R	422419104007
BALAMURUGAN M	422419104009
SATHYANARAYANAN S	422419104036

TEAM ID: PNT2022TMID39159

University college of Engineering Tindivanam

TABLE OF CONTENTS:

1. 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

The Purpose of our Project is

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

2. LITERATURE SURVEY

2.1 Existing Problem :

> In our society, we have people with disabilities.

> The technology is developing day by day but no significant developments are undertaken for the betterment of these people.

> Communications between deaf-mute and a normal person has always been a challenging task.

> It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language.

2.2 References :

☆https://www.researchgate.net/publication/228618921_A_literature_review_on_artificial_intelligence

☆ https://journals.sagepub.com/doi/full/10.1177/02683962211048201

2.3 Problem Statement definition :

> People with disabilities are having a difficult time keeping up with the rapidly evolving technology, which is one of the major issues that our society is dealing with.

> For those with disabilities, having access to communication tools has become crucial.

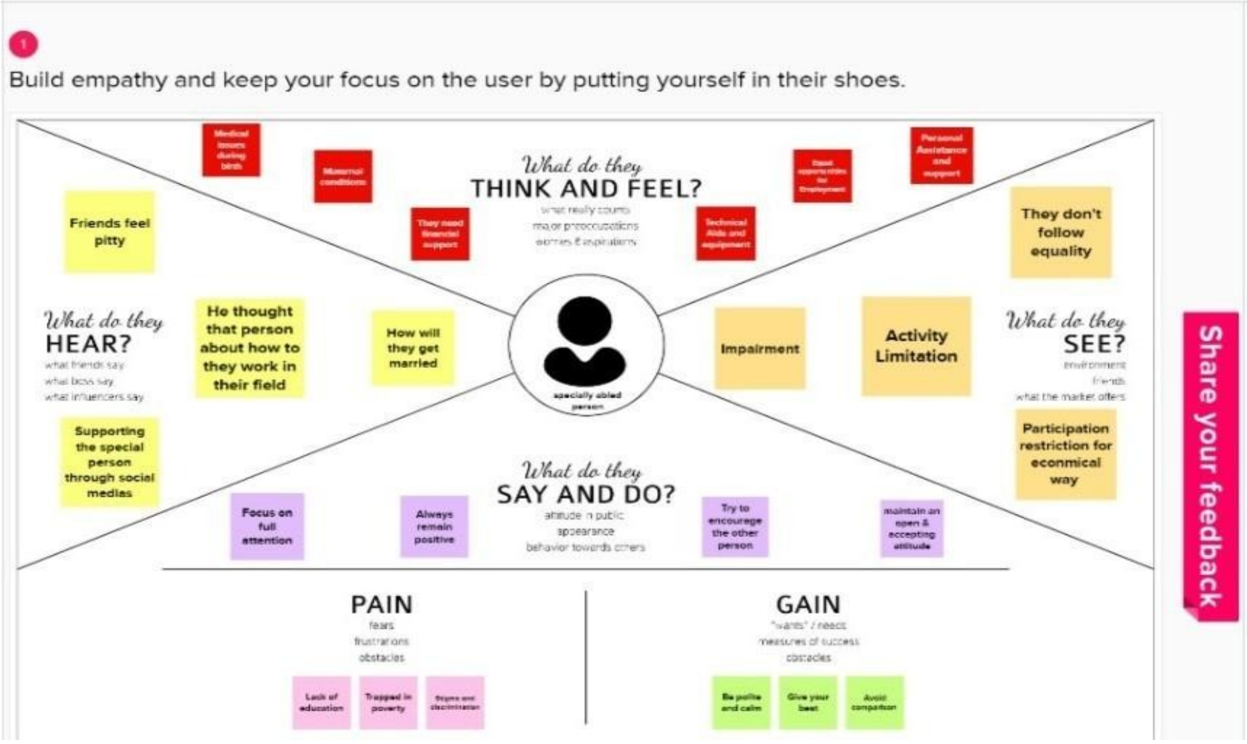
> Typically deaf and stupid people use sign language to communicate, but they struggle to do so with non-sign language users language.

> Information is the main topic of communication between normal and deaf individuals using sign language which is expressive and natural.

> So that we can converse with them and comprehend what they're saying, we need a translation. A language translation technology converts common sign language into voice, enabling regular people to communicate with one another.

3. IDEATION & PROPOSED SOLUTION

3.1EMPATHY MAP CANVAS



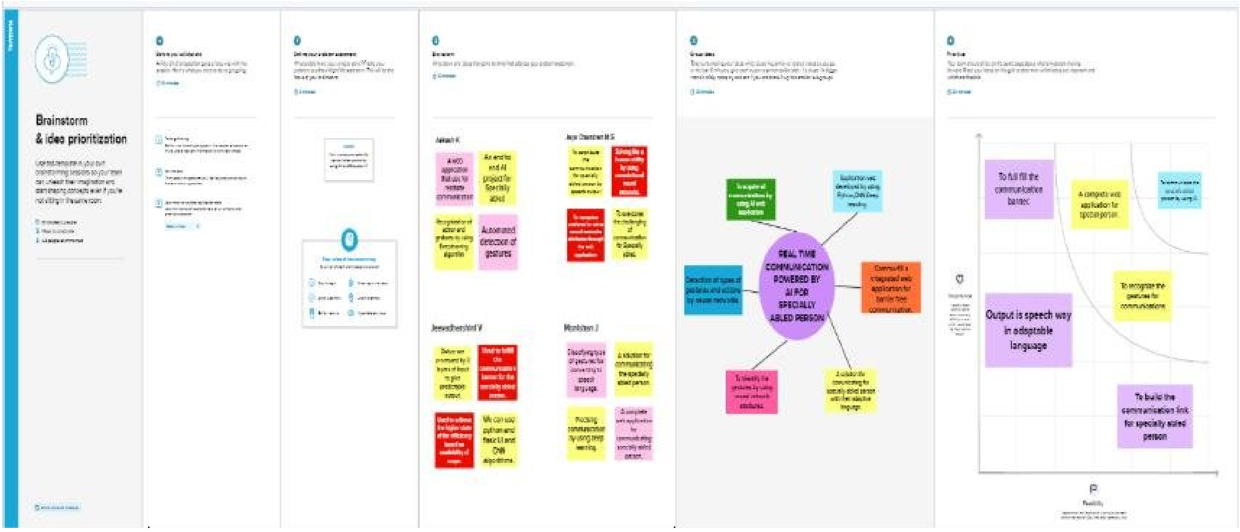
1

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

Link:

☆ https://app.mural.co/invitation/mural/realtimecommunicationpowered2234/1666079639406?sender=u213ba3fcfda4b22518119207&key=51fe48f5-c5b8-4348-a041-b7971de518f4

3.2 Ideation & Brainstorming


Brainstorm
& idea prioritization
REAL TIME COMMUNICATION POWERED BY AI FOR SPECIALLY ABLED PERSON
Automated detection of gestures
We can use python and flask UI and CNN algorithms.
To full fill the communication barrier.
A complete web application for special person.
To recognize the gestures for communications
Output is speech way in adaptable language
To build the communication link for specially abled person

Link:

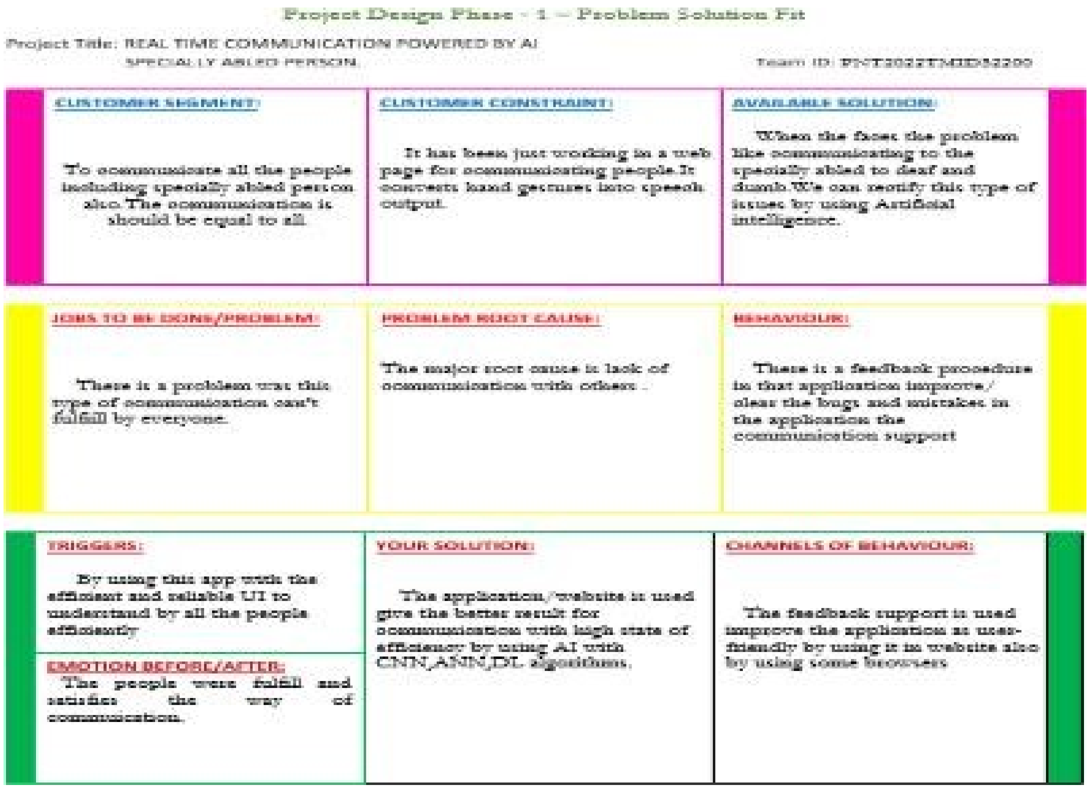
☆ https://app.mural.co/invitation/mural/realtimecommunicationpowered2234/1666079639406?sender=u213ba3fcfda4b22518119207&key=51fe48f5-c5b8-4348-a041-b7971de518f4

3.3 Proposed Solution

S.No .	Parameter	Description
1	Problem Statement (Problem to be solved)	● 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.
2	Idea / Solution description	● By using Artificial intelligence to convert hand gestures to speech output. ● To desire the fulfillment of communication.

3	Novelty / Uniqueness	● User friendly UI to access the web application by all the people. ● It was access by everyone.
4	Social Impact / Customer Satisfaction	● It Improves level of the communication. ● It will provide the better way to communication.
5	Business Model (Revenue Model)	● Using the cloud storage, we can easily access the web application. ● We can use anytime and anywhere.
6	Scalability of the Solution	★ Increasing the scalability of to achieve higher state of efficiency.

3.4 Problem Solution fit



Project Design Phase - 1 – Problem Solution Fit

Project Title: REAL TIME COMMUNICATION POWERED BY AI SPECIALLY ABLED PERSON

Team ID: PNT2022TMID52200

CUSTOMER SEGMENT:

To communicate all the people including specially abled person also.The communication is should be equal to all.

CUSTOMER CONSTRAINT:

It has been just working in a web page for communicating people.It converts hand gestures into speech output.

AVAILABLE SOLUTION:

When the faces the problem like communicating to the specially abled to deaf and dumb.We can rectify this type of issues by using Artificial intelligence.

JOBS TO BE DONE/PROBLEM:

There is a problem was this type of communication can't fulfill by everyone.

PROBLEM ROOT CAUSE:

The major root cause is lack of communication with others .

BEHAVIOUR:

There is a feedback procedure in that application improve/ clear the bugs and mistakes in the application the communication support

TRIGGERS:

By using this app with the efficient and reliable UI to understand by all the people efficiently

EMOTION BEFORE/AFTER:

The people were fulfill and satisfies the way of communication.

YOUR SOLUTION:

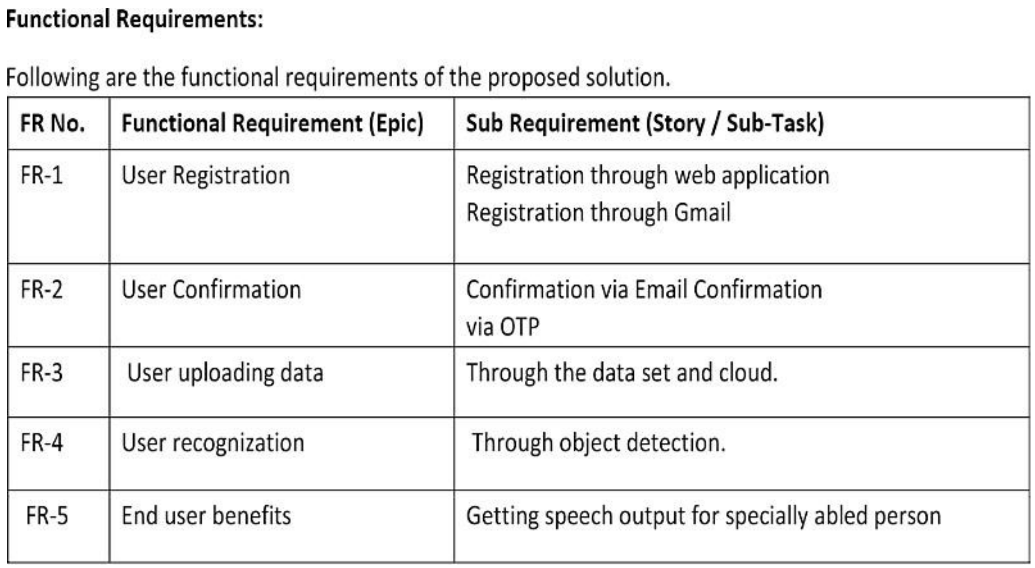
The application/website is used give the better result for communication with high state of efficiency by using AI with CNN,ANN,DL algorithms.

CHANNELS OF BEHAVIOUR:

The feedback support is used improve the application as user-friendly by using it in website also by using some browsers

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

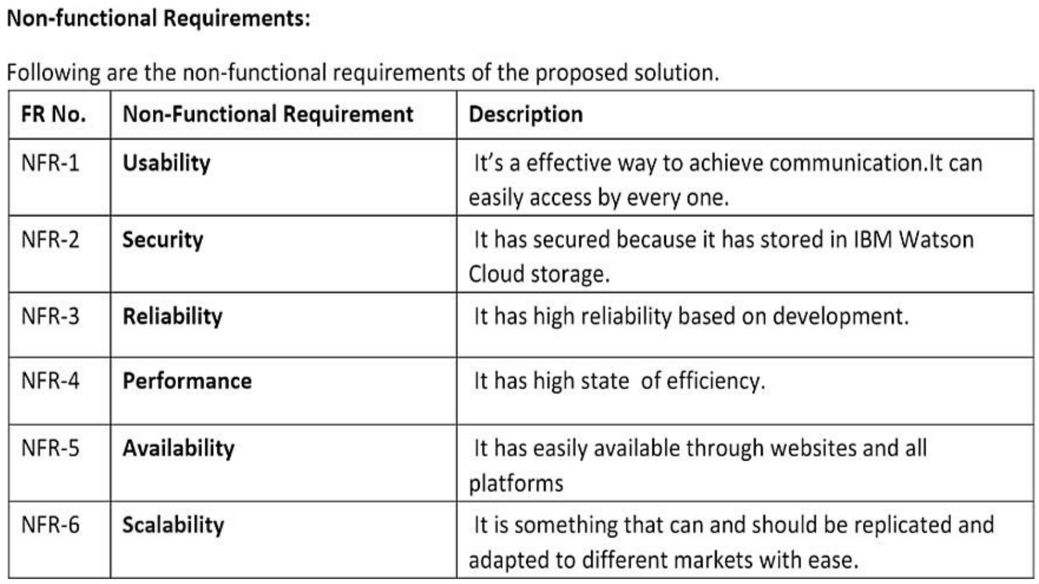


Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through web application Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User uploading data	Through the data set and cloud.
FR-4	User recognization	Through object detection.
FR-5	End user benefits	Getting speech output for specially abled person

4.2 Non Functional requirements



Non-functional Requirements:

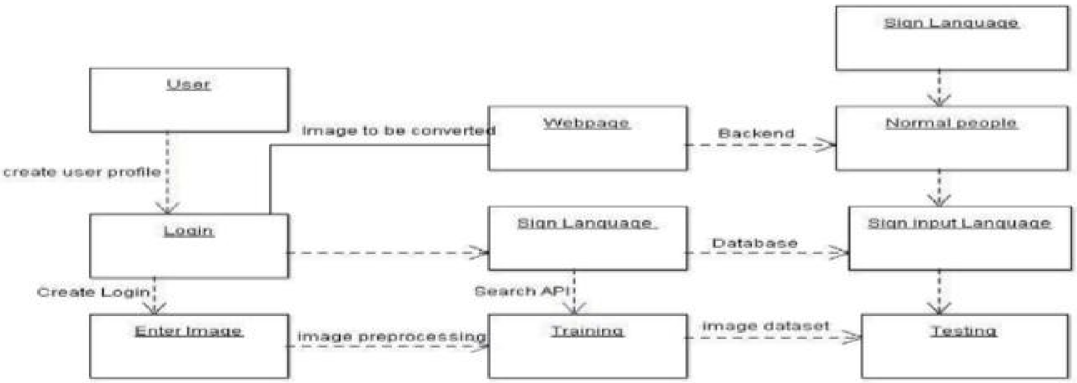
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It's a effective way to achieve communication.It can easily access by every one.
NFR-2	**Security**	It has secured because it has stored in IBM Watson Cloud storage.
NFR-3	**Reliability**	It has high reliability based on development.
NFR-4	**Performance**	It has high state of efficiency.
NFR-5	**Availability**	It has easily available through websites and all platforms
NFR-6	**Scalability**	It is something that can and should be replicated and adapted to different markets with ease.

5. PROJECT DESIGN

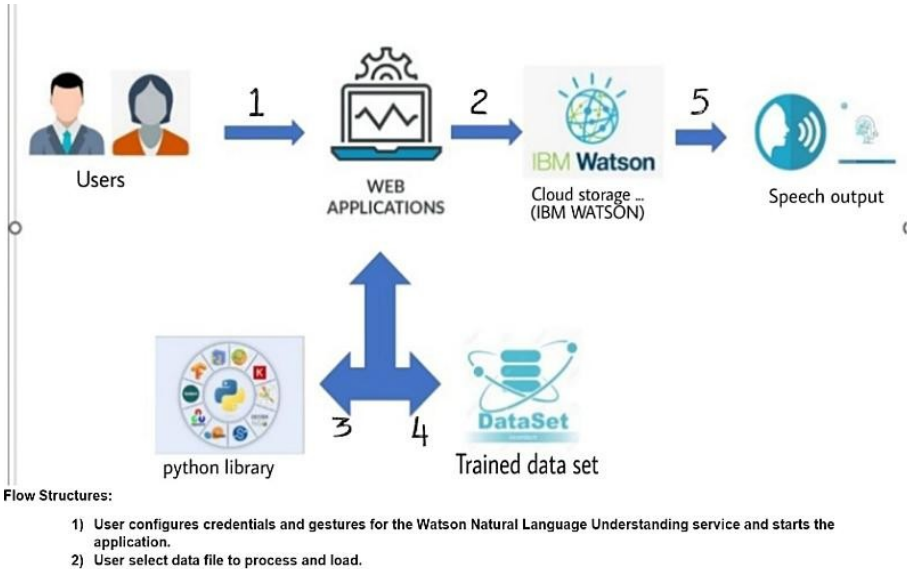
5.1 Data Flow Diagram

Example: Data Flow diagram



5.2 Solution and Technical Architecture

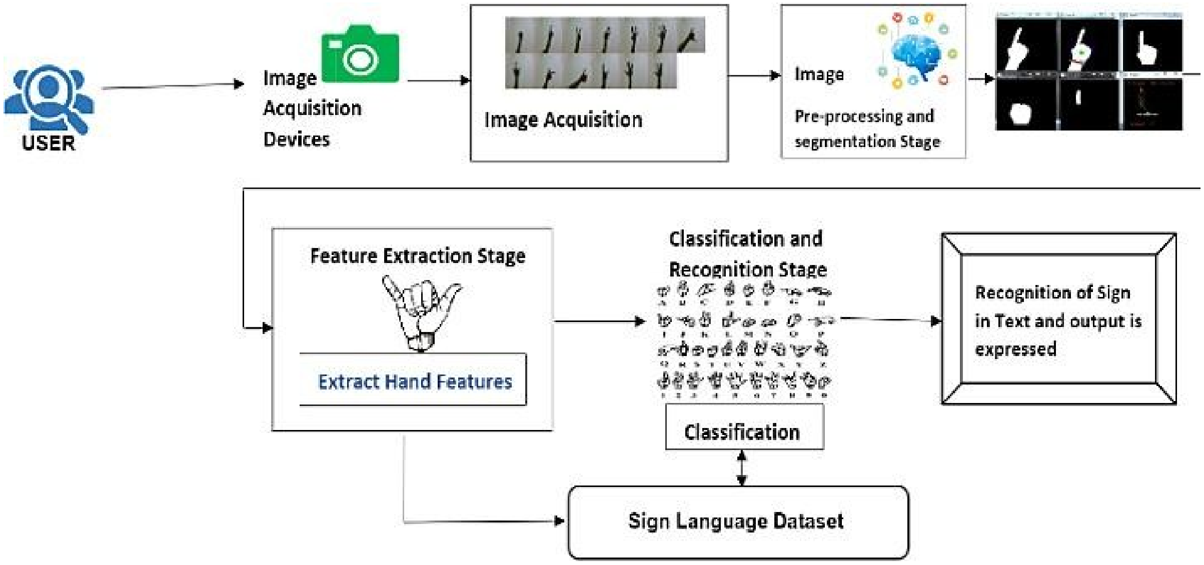
Solution Architecture:



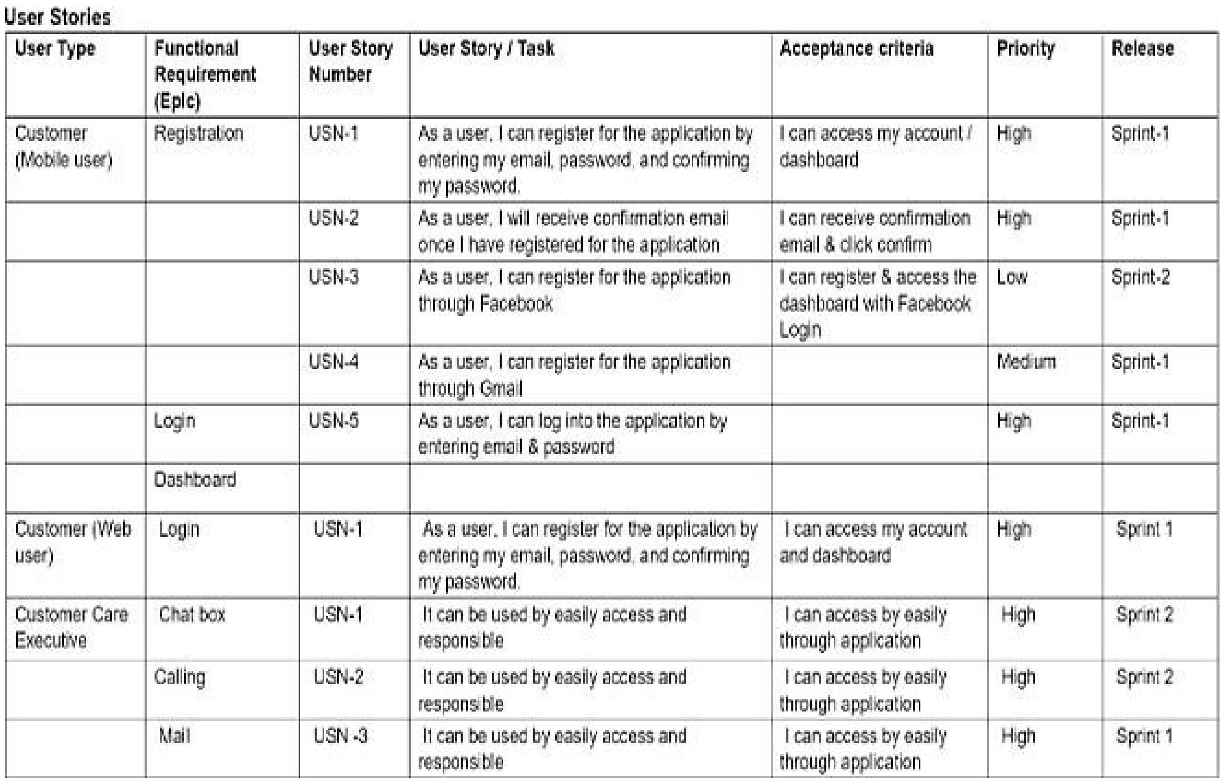
Flow Structures:

1) User configures credentials and gestures for the Watson Natural Language Understanding service and starts the application.
2) User select data file to process and load.

Technical Architecture:


USER
Image Acquisition Devices
Image Acquisition
Image
Pre-processing and segmentation Stage
Feature Extraction Stage
Extract Hand Features
Classification and Recognition Stage
Classification
Recognition of Sign in Text and output is expressed
Sign Language Dataset

5.3 User Stories



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)	Login	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account and dashboard	High	Sprint 1
Customer Care Executive	Chat box	USN-1	It can be used by easily access and responsible	I can access by easily through application	High	Sprint 2
	Calling	USN-2	It can be used by easily access and responsible	I can access by easily through application	High	Sprint 2
	Mail	USN -3	It can be used by easily access and responsible	I can access by easily through application	High	Sprint 1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
SPRINT 1					
Sprint-1	Data Collection	USN-1	Collect Dataset	9	High
Sprint-1		USN-2	Image Preprocessing	7	Medium
SPRINT 2					

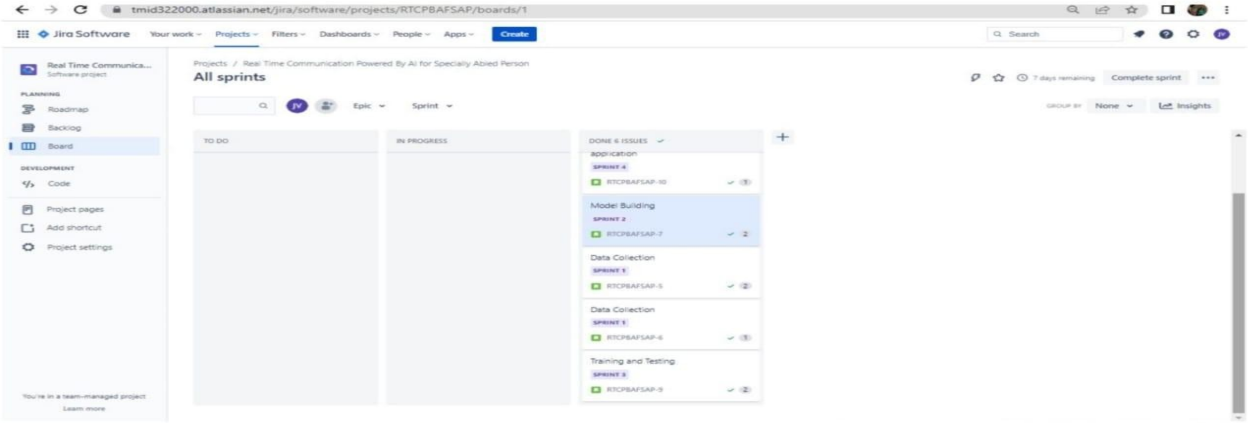
Sprint-2	Model Building	USN-3	Image the required libraries, add the necessary layers and compile the model.	10	High
Sprint-2	Dashboard	USN-4	Training the image classification model using CNN	8	Medium

SPRINT 3					
Sprint-3	Training and Testing	USN-5	Training the model and testing the model's performance	9	High
SPRINT 4					
Sprint-4	Implementation of the application	USN-6	Converting the input sign language images into English alphabets	8	Medium

6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	04 Nov 2022	8	29 Oct 2022
Sprint-2	20	6 Days	01 Nov 2022	08 Nov 2022	5	05 Nov 2022
Sprint-3	20	6 Days	08 Nov 2022	12 Nov 2022	7	12 Nov 2022
Sprint-4	20	6 Days	15 Nov 2022	16 Nov 2022	5	18 Nov 2022

6.3 Reports from JIRA:



Projects / Real Time Communication Powered By AI for Specially Abled Person

All sprints

TO DO

IN PROGRESS

DONE 6 ISSUES

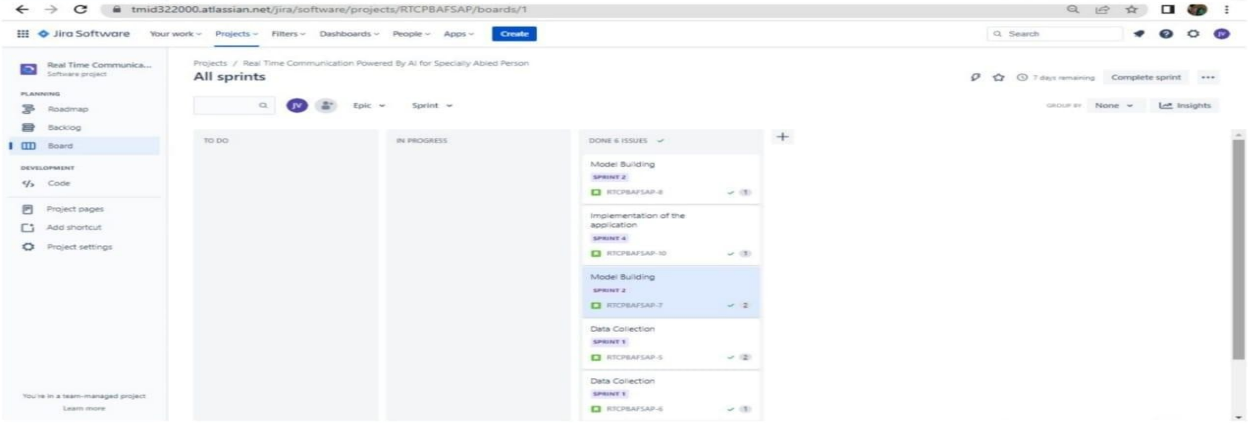
application
SPRINT 4
RTCPBAFSAP-10 ✓ 1

Model Building
SPRINT 2
RTCPBAFSAP-7 ✓ 2

Data Collection
SPRINT 1
RTCPBAFSAP-5 ✓ 2

Data Collection
SPRINT 1
RTCPBAFSAP-6 ✓ 1

Training and Testing
SPRINT 3
RTCPBAFSAP-9 ✓ 2



Projects / Real Time Communication Powered By AI for Specially Abled Person

All sprints

TO DO

IN PROGRESS

DONE 6 ISSUES

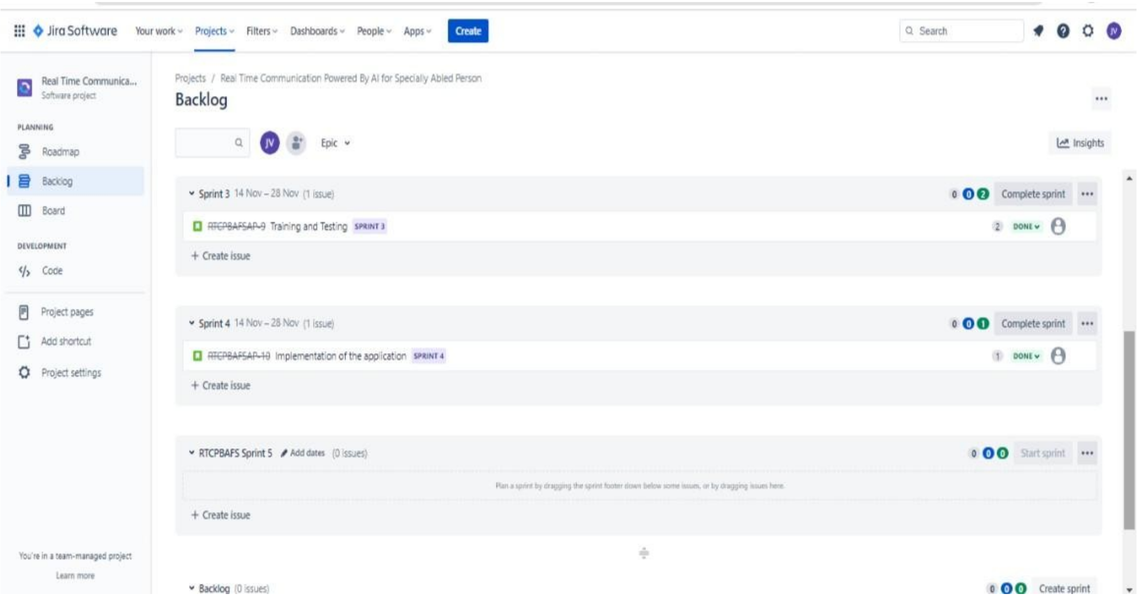
Model Building
SPRINT 2
RTCPBAFSAP-8 ✓ 1

Implementation of the application
SPRINT 4
RTCPBAFSAP-10 ✓ 1

Model Building
SPRINT 2
RTCPBAFSAP-7 ✓ 2

Data Collection
SPRINT 1
RTCPBAFSAP-5 ✓ 2

Data Collection
SPRINT 1
RTCPBAFSAP-6 ✓ 1



Projects / Real Time Communication Powered By AI for Specially Abled Person

Backlog

Sprint 3 14 Nov – 28 Nov (1 issue) 0 0 2 Complete sprint

~~RTCPBAFSAP-9~~ Training and Testing SPRINT 3 2 DONE

+ Create issue

Sprint 4 14 Nov – 28 Nov (1 issue) 0 0 1 Complete sprint

~~RTCPBAFSAP-10~~ Implementation of the application SPRINT 4 1 DONE

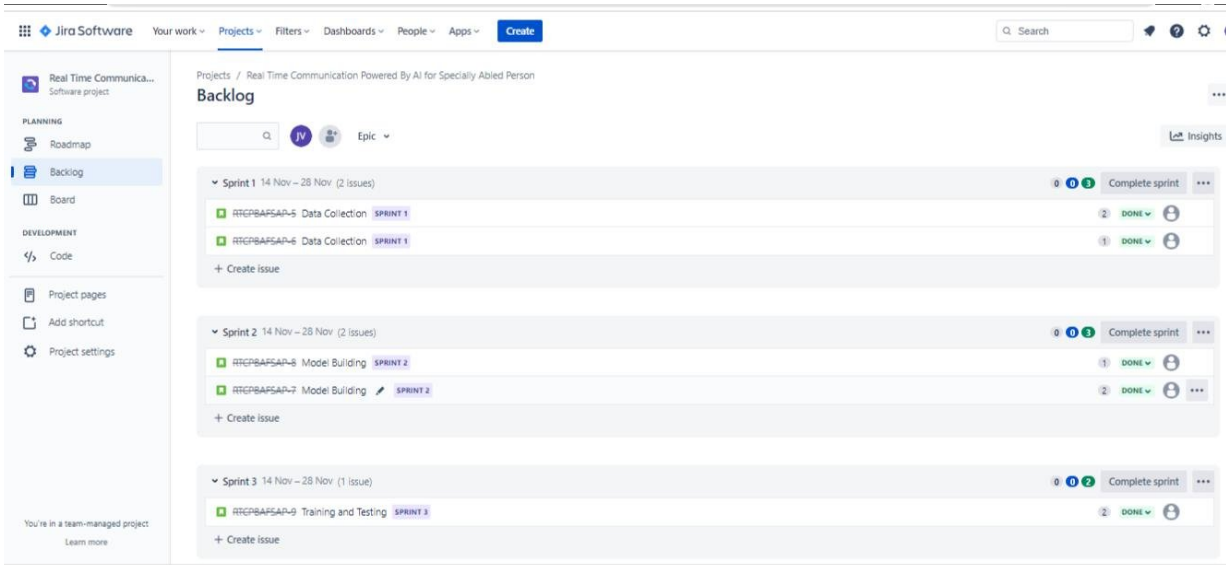
+ Create issue

RTCPBAFS Sprint 5 Add dates (0 issues) 0 0 0 Start sprint

Plan a sprint by dragging the sprint footer down below some issues, or by dragging issues here.

+ Create issue

Backlog (0 issues) 0 0 0 Create sprint



Projects / Real Time Communication Powered By AI for Specially Abled Person

Backlog

Sprint 1 14 Nov – 28 Nov (2 issues) 0 0 3 Complete sprint

~~RTCPBAFSAP-5~~ Data Collection	SPRINT 1	2	DONE	
~~RTCPBAFSAP-6~~ Data Collection	SPRINT 1	1	DONE	

+ Create issue

Sprint 2 14 Nov – 28 Nov (2 issues) 0 0 3 Complete sprint

~~RTCPBAFSAP-8~~ Model Building	SPRINT 2	1	DONE	
~~RTCPBAFSAP-7~~ Model Building	SPRINT 2	2	DONE	

+ Create issue

Sprint 3 14 Nov – 28 Nov (1 issue) 0 0 2 Complete sprint

~~RTCPBAFSAP-9~~ Training and Testing	SPRINT 3	2	DONE	

+ Create issue

7. CODING & SOLUTIONING

7.1 Feature 1

Description:

Recognizing the ASL

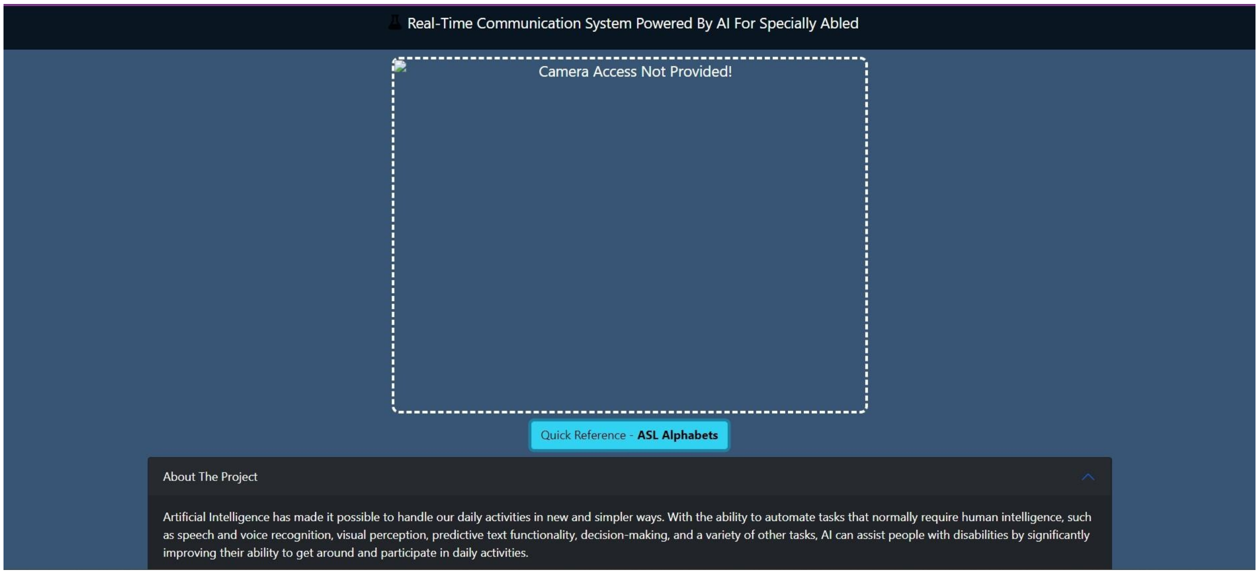
In feature 1 we have designed a webpage using flask app to recognize the hand gestures to communicate the specially abled person.The first part of this project is interface that interface was designed by html and CSS tools for designing and We have inserted the reference sign letters from ASL (American Sign Language).

HTML code:

```
 <!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>RTC_VideoTemplate</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.12.0/css/all.css">
    <link rel="stylesheet" href="assets/css/Banner-Heading-Image.css">
    <link rel="stylesheet" href="assets/css/Navbar-Centered-Brand.css">
    <link rel="stylesheet" href="assets/css/styles.css">
</head>

<body style="background: rgb(57, 85, 117);">
    <nav class="navbar navbar-light navbar-expand-md py-3" style="background: #091520;">
        <div class="container">
            <div></div><a class="navbar-brand d-flex align-items-center" href="#"><span
                    class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex justify-content-center
align-items-center me-2 bs-icon"><i
                        class="fas fa-flask"></i></span><span style="color: rgb(255,255,255);">Real-
Time Communication
                    System Powered By AI&nbsp;For Specially Abled</span></a>
            <div></div>
        </div>
    </nav>
    <section>
        <div class="d-flex flex-column justify-content-center align-items-center">
            <div class="d-flex flex-column justify-content-center align-items-center" id="div-video-
feed"
                style="width: 640px;height: 480px;margin: 10px;min-height: 480px;min-width:
640px;border-radius: 10px;border: 4px dashed rgb(255,255,255) ;">
                <img src="{{ url_for('video_feed') }}" style="width: 100%;height: 100%;color:
rgb(255,255,255);text-align: center;font-size: 20px;"
                    alt="Camera Access Not Provided!">
            </div>
        </div>
        <div class="d-flex flex-column justify-content-center align-items-center" style="margin-
bottom: 10px;"><button
                class="btn btn-info" type="button" data-bs-target="#modal-1" data-bs-
toggle="modal">Quick Reference
                -<strong> ASL Alphabets</strong></button></div>
    </section>
    <section>
        <div class="container">
            <div class="accordion text-white" role="tablist" id="accordion-1">
                <div class="accordion-item" style="background: rgb(33,37,41);">
```

```
                    <h2 class="accordion-header" role="tab"><button class="accordion-button" data-
bs-toggle="collapse"
                            data-bs-target="#accordion-1 .item-1" aria-expanded="true"
                            aria-controls="accordion-1 .item-1"
                            style="background: rgb(39,43,48);color: rgb(255,255,255);">About The
Project</button></h2>
                    <div class="accordion-collapse collapse show item-1" role="tabpanel" data-bs-
parent="#accordion-1">
                        <div class="accordion-body">
                            <p class="mb-0">Artificial Intelligence has made it possible to handle our
daily activities
                                in new and simpler ways. With the ability to automate tasks that normally
require human
                                intelligence, such as speech and voice recognition, visual perception,
predictive text
                                functionality, decision-making, and a variety of other tasks, AI can assist
people with
                                disabilities by significantly improving their ability to get around and
participate in
                                daily activities.<br><br>Currently, Sign Recognition is available
<strong>only for
                                    alphabets A-I</strong> and not for J-Z, since J-Z alphabets also require
Gesture
                                Recognition for them to be able to be predicted correctly to a certain degree
of
                                accuracy.</p>
                        </div>
                    </div>
                </div>
                <div class="accordion-item" style="background: rgb(33,37,41);">
                    <h2 class="accordion-header" role="tab"><button class="accordion-button
collapsed"
                            data-bs-toggle="collapse" data-bs-target="#accordion-1 .item-2" aria-
expanded="false"
                            aria-controls="accordion-1 .item-2"
                            style="background: rgb(39,43,48);color: rgb(237, 194, 38);">Developed
By</button></h2>
```

```
                    <div class="accordion-collapse collapse item-2" role="tabpanel" data-bs-
parent="#accordion-1">
                        <div class="accordion-body">
                            <p class="mb-0">Students at Nandha college of technology doing for nalaiya
thiran
                                Program.<br><br>1. <strong>AAKASH</strong> 732119106001<br>2.
                                <strong>JAI</strong> 732119106014<br>3.
<strong>MONISH</strong>732119106023<br>4.<strong>
JEEVA</strong>732119106015</br>.
                            </p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>
    <div class="modal fade" role="dialog" tabindex="-1" id="modal-1">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h4 class="modal-title">American Sign Language - Alphabets</h4><button
type="button"
                        class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body"><img src="{{ url_for('static',
filename='img/ASL_Alphabets.png') }}" width="100%"></div>
                <div class="modal-footer"><button class="btn btn-secondary" type="button"
                        data-bs-dismiss="modal">Close</button></div>
            </div>
        </div>
    </div>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>

</html>
```



Real-Time Communication System Powered By AI For Specially Abled

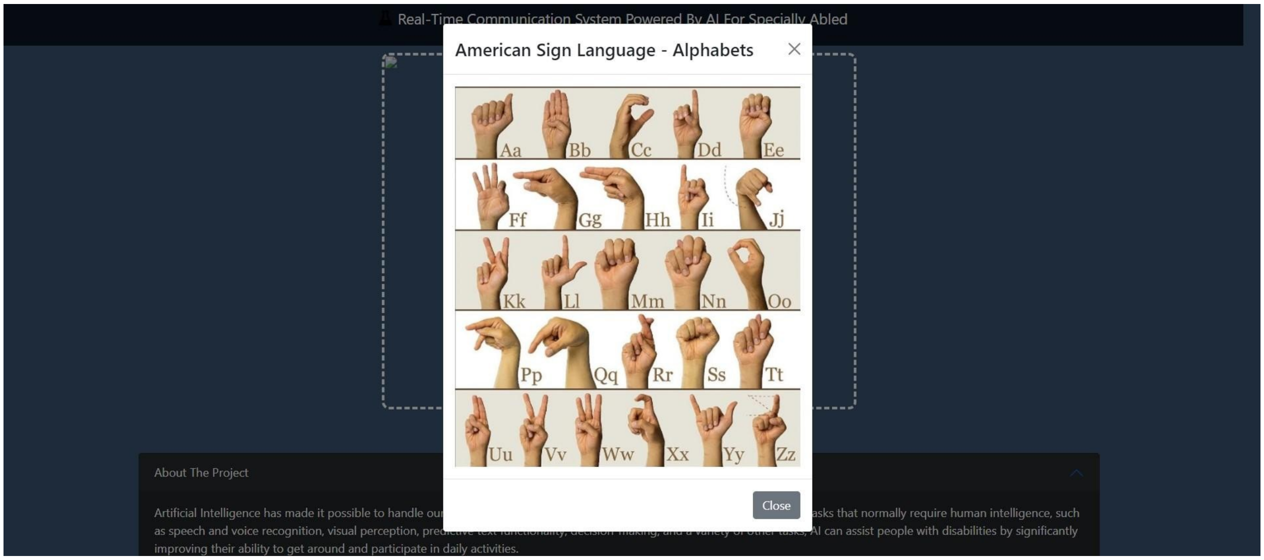
Camera Access Not Provided!

About The Project

Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly improving their ability to get around and participate in daily activities.

We inserted the Quick reference-ASL Alphabets for understanding the sign letters

American Sign Language (ASL) is a complete, natural language that has the same linguistic properties as spoken languages, with grammar that differs from English. ASL is expressed by movements of the hands and face. It is the primary language of many North Americans who are deaf and hard of hearing and is used by some hearing people as well.



Real-Time Communication System Powered By AI For Specially Abled

American Sign Language - Alphabets

About The Project

Artificial Intelligence has made it possible to handle our ... asks that normally require human intelligence, such as speech and voice recognition, visual perception, pre... AI can assist people with disabilities by significantly improving their ability to get around and participate in daily activities.

7.1 Feature 2

Recognizing the hand gestures to the speeech output

- Import necessary packages.
- Initialize models.
- Read frames from a webcam.
- Detect hand keypoints.
- Recognize hand gestures.

Step 1 – Import necessary packages:

To build this Hand Gesture Recognition project, we’ll need four packages. So first import these.

```
# import necessary packages for hand gesture recognition project using Python OpenCV

import cv2
import numpy as np
import mediapipe as mp
import tensorflow as tf
from tensorflow.keras.models import load_model
```

Step 2 – Initialize models:

```
# initialize mediapipe
mpHands = mp.solutions.hands
hands = mpHands.Hands(max_num_hands=1, min_detection_confidence=0.7)
mpDraw = mp.solutions.drawing_utils
```

Step 3 – Read frames from a webcam:

```
# Initialize the webcam for Hand Gesture Recognition Python project
cap = cv2.VideoCapture(0)

while True:
  # Read each frame from the webcam
  _, frame = cap.read()
x , y, c = frame.shape

  # Flip the frame vertically
  frame = cv2.flip(frame, 1)
  # Show the final output
  cv2.imshow("Output", frame)
  if cv2.waitKey(1) == ord('q'):
        break

# release the webcam and destroy all active windows
cap.release()
cv2.destroyAllWindows()
```

- We create a VideoCapture object and pass an argument ‘0’. It is the camera ID of the system. In this case, we have 1 webcam connected with the system. If you have multiple webcams then change the argument according to your camera ID. Otherwise, leave it default.
- The cap.read() function reads each frame from the webcam.
- cv2.flip() function flips the frame.
- cv2.imshow() shows frame on a new openCV window.
- The cv2.waitKey() function keeps the window open until the key ‘q’ is pressed.

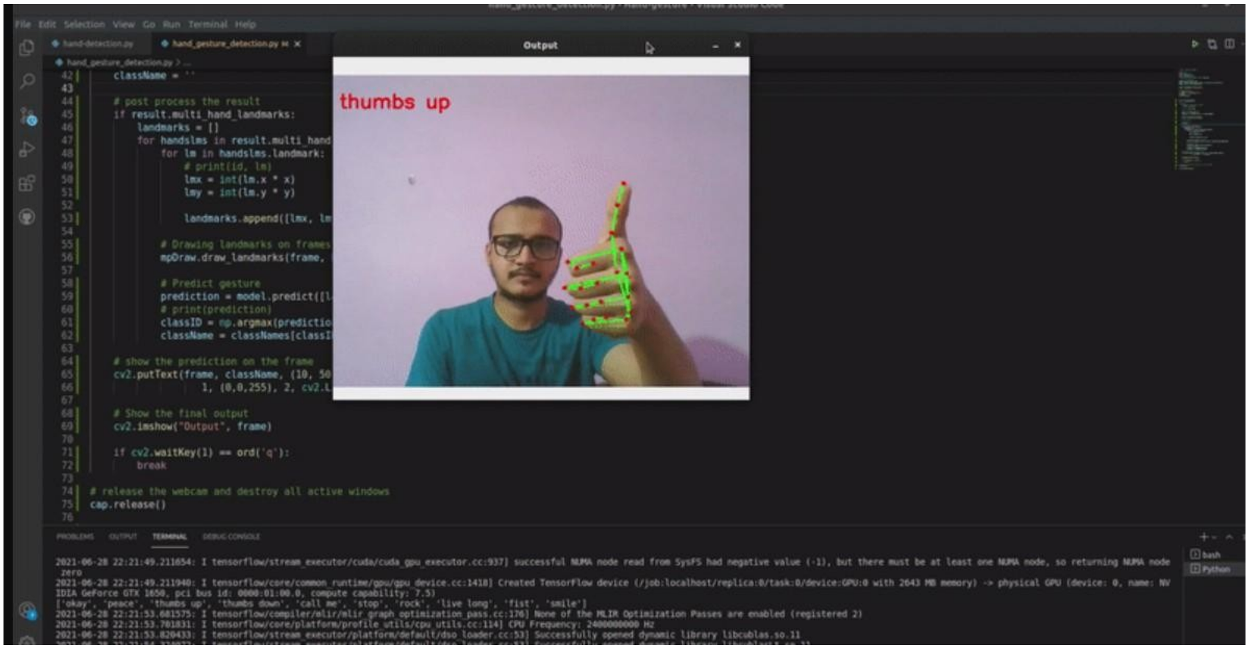
Step 4 – Detect hand keypoints:

```
framergb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
  # Get hand landmark prediction
  result = hands.process(framergb)


  className = ''


  # post process the result
  if result.multi_hand_landmarks:
      landmarks = []
      for handslms in result.multi_hand_landmarks:
          for lm in handslms.landmark:
              # print(id, lm)
              lmx = int(lm.x * x)
              lmy = int(lm.y * y)


              landmarks.append([lmx, lmy])
```

Step 5 – Recognize hand gestures:

```
    # Predict gesture in Hand Gesture Recognition
        projectprediction = model.predict([landmarks])
print(prediction)
        classID = np.argmax(prediction)
        className = classNames[classID]

  # show the prediction on the frame
  cv2.putText(frame, className, (10, 50), cv2.FONT_HERSHEY_SIMPLEX,
              1, (0,0,255), 2, cv2.LINE_AA)
```

OUTPUT:



File Edit Selection View Go Run Terminal Help
hand-detection.py
hand_gesture_detection.py
Output
thumbs up
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
bash
Python

```
        className = ''

        # post process the result
        if result.multi_hand_landmarks:
            landmarks = []
            for handslms in result.multi_hand
                for lm in handslms.landmark:
                    # print(id, lm)
                    lmx = int(lm.x * x)
                    lmy = int(lm.y * y)

                    landmarks.append([lmx, lm

                # Drawing landmarks on frames
                mpDraw.draw_landmarks(frame,

                # Predict gesture
                prediction = model.predict([[l
                # print(prediction)
                classID = np.argmax(predictio
                className = classNames[classI

        # show the prediction on the frame
        cv2.putText(frame, className, (10, 50
                    1, (0,0,255), 2, cv2.L

        # Show the final output
        cv2.imshow("Output", frame)

        if cv2.waitKey(1) == ord('q'):
            break

    # release the webcam and destroy all active windows
    cap.release()
```

```
2021-06-28 22:21:49.211654: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node
 zero
2021-06-28 22:21:49.211940: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1418] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 2643 MB memory) -> physical GPU (device: 0, name: NV
IDIA GeForce GTX 1650, pci bus id: 0000:01:00.0, compute capability: 7.5)
['okay', 'peace', 'thumbs up', 'thumbs down', 'call me', 'stop', 'rock', 'live long', 'fist', 'smile']
2021-06-28 22:21:53.681575: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:176] None of the MLIR Optimization Passes are enabled (registered 2)
2021-06-28 22:21:53.701831: I tensorflow/core/platform/profile_utils/cpu_utils.cc:114] CPU Frequency: 2400000000 Hz
2021-06-28 22:21:53.820433: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcublas.so.11
```

Coding for web application building

Python flask application

```
from flask import Flask, Response, render_template
from camera import Video

app = Flask(_name_)
@app.route('/')
def index():
        return render_template('index.html')

def gen(camera):
        while True:
                frame = camera.get_frame()
                yield(b'--frame\r\n'
                        b'Content-Type: image/jpeg\r\n\r\n' + frame +
                        b'\r\n\r\n')

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


if__name__== ' main ':
        app.run()
```

Code for accesing camera

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

class Video(object):
        def__init_(self):
                self.video = cv2.VideoCapture(0)
                self.roi_start = (50, 150)
                self.roi_end = (250, 350)
                self.model = load_model('IBM_Communication_Model.h5') # Execute IBM
Trained Model
```

```
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def___del_(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is:
'+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

Main code (video accessing)

```
import cv2

video = cv2.VideoCapture(0)

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

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

LOADING THE DATA SET

```
# Training Datagen
train_datagen =
ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

MODEL BUILDING

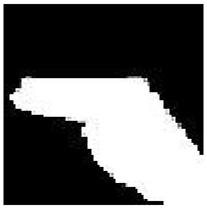
```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
  "colab": {
   "provenance": []
  },
  "kernelspec":
   { "name":
   "python3",
   "display_name": "Python 3"
  },
  "language_info":
   {"name":
   "python"
  }
 },
 "cells": [
  {
   "cell_type": "markdown",
   "source": [
    "# **Real-Time Communication System Powered By AI For Specially Abled**"
   ],
   "metadata": {
    "id": "FAGzis88l7x8"
   }
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Loading the Dataset & Image Data Generation**"
```

],

```
    "metadata": {
     "id": "c3y70G0zl7Yn"
    }
   },
   {
    "cell_type": "code",
    "execution_count": 3,
    "metadata": {
     "id": "iy2QXRwJeOqr"
    },
    "outputs": [],
    "source": [
     "from tensorflow.keras.preprocessing.image import ImageDataGenerator"
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "# Training Datagen\n",
     "train_datagen =
ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)\n
",
     "# Testing Datagen\n",
     "test_datagen = ImageDataGenerator(rescale=1/255)\n"
    ],
    "metadata": {
     "id": "AdFUXM70fmPj"
    },
    "execution_count": 4,
    "outputs": []
   },
   {
    "cell_type": "code",
    "source": [
     "# Training Dataset\n",

"x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',targe
t_size=(64,64), class_mode='categorical',batch_size=900)\n",
    "# Testing Dataset\n",

"x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size
=(64,64), class_mode='categorical',batch_size=900)\n"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "-SpHowmAgu7_",
    "outputId": "a456a77d-b9dc-47cb-b942-67453f1a81b7"
   },
   "execution_count": 6,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout", "text":
     [
      "Found 15760 images belonging to 9 classes.\n",
      "Found 2250 images belonging to 9 classes.\n"
     ]
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "print(\"Len x-train : \", len(x_train))\n",
    "print(\"Len x-test : \", len(x_test))"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "2qLcDqP4jgPT",
    "outputId": "4bf5a506-506c-44ac-9f13-040570b3643a"
  },
  "execution_count": 7,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout", "text":
      [
        "Len x-train : 18\n",
        "Len x-test : 3\n"
      ]
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "# The Class Indices in Training Dataset\n",
    "x_train.class_indices"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "V9Z-Rvl1jh-Q",
    "outputId": "d67bde72-545f-4820-84f5-2885822f7c10"
  },
  "execution_count": 8,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}"
        ]
      },
      "metadata": {},
      "execution_count": 8
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Model Creation**"
   ],
   "metadata": {
    "id": "5yHOh0Bhl5F9"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "# Importing Libraries\n",
    "from tensorflow.keras.models import Sequential\n",
    "from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense"
   ],
   "metadata": {
    "id": "ycQhnJ3om87I"
   },
   "execution_count": 9,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "# Creating Model\n",
    "model=Sequential()"
   ],
   "metadata": {
    "id": "IVNzGYblocSh"
   },
   "execution_count": 10,
   "outputs": []
  },
{
 "cell_type": "code",
 "source": [
  "# Adding Layers\n",
  "model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))"
 ],
 "metadata": {
  "id": "G7kEjSISpDs7"
 },
 "execution_count": 11,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "model.add(MaxPooling2D(pool_size=(2,2)))"
 ],
 "metadata": {
  "id": "p8lwdE26pLdN"
 },
 "execution_count": 12,
 "outputs": []
},
{
 "cell_type": "code",
 "source":
 [ "model.add(Flatten())"
 ],
 "metadata": {
  "id": "cIeLXS77pTEq"
 },
 "execution_count": 13,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
    "# Adding Dense Layers\n",
    "model.add(Dense(300,activation='relu'))\n",
    "model.add(Dense(150,activation='relu'))\n",
    "model.add(Dense(9,activation='softmax'))"
   ],
   "metadata": {
    "id": "0XAR5Q0fphqp"
   },
   "execution_count": 14,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "# Compiling the Model\n",
    "model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])"
   ],
   "metadata": {
    "id": "Pvo6cZAVpsiT"
   },
   "execution_count": 15,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "# Fitting the Model Generator\n",
    "model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "a1tPmi7ap5yd",
    "outputId": "fe240748-001e-43ab-f0cc-3e955747250a"
    },
    "execution_count": 18,
    "outputs": [
     {
      "output_type": "stream",
      "name": "stderr",
      "text": [
       "/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version. Please use
`Model.fit`, which supports generators.\n",
       " \n"
      ]
     },
     {
      "output_type": "stream",
      "name": "stdout", "text":
      [
       "Epoch 1/10\n",
       "18/18 [==============================] - 92s 5s/step - loss: 0.0049 -
accuracy: 0.9994 - val_loss: 0.2635 - val_accuracy: 0.9773\n",
       "Epoch 2/10\n",
       "18/18 [==============================] - 90s 5s/step - loss: 0.0040 -
accuracy: 0.9995 - val_loss: 0.2074 - val_accuracy: 0.9773\n",
       "Epoch 3/10\n",
       "18/18 [==============================] - 87s 5s/step - loss: 0.0041 -
accuracy: 0.9995 - val_loss: 0.2460 - val_accuracy: 0.9773\n",
       "Epoch 4/10\n",
       "18/18 [==============================] - 91s 5s/step - loss: 0.0041 -
accuracy: 0.9992 - val_loss: 0.2470 - val_accuracy: 0.9782\n",
       "Epoch 5/10\n",
       "18/18 [==============================] - 88s 5s/step - loss: 0.0037 -
accuracy: 0.9993 - val_loss: 0.2439 - val_accuracy: 0.9782\n",
       "Epoch 6/10\n",
       "18/18 [==============================] - 88s 5s/step - loss: 0.0024 -
accuracy: 0.9997 - val_loss: 0.2852 - val_accuracy: 0.9782\n",
       "Epoch 7/10\n",
       "18/18 [==============================] - 91s 5s/step - loss: 0.0023 -
accuracy: 0.9997 - val_loss: 0.2589 - val_accuracy: 0.9782\n",
          "Epoch 8/10\n",
          "18/18 [==============================] - 93s 5s/step - loss: 0.0014 -
accuracy: 1.0000 - val_loss: 0.2523 - val_accuracy: 0.9782\n",
          "Epoch 9/10\n",
          "18/18 [==============================] - 92s 5s/step - loss: 0.0013 -
accuracy: 0.9999 - val_loss: 0.2269 - val_accuracy: 0.9778\n",
          "Epoch 10/10\n",
          "18/18 [==============================] - 91s 5s/step - loss: 0.0012 -
accuracy: 0.9999 - val_loss: 0.2968 - val_accuracy: 0.9782\n"
        ]
      },
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "<keras.callbacks.History at 0x7fde26f54590>"
          ]
        },
        "metadata": {},
        "execution_count": 18
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "**Saving the Model**"
    ],
    "metadata": {
      "id": "jB3lpZWWIBi9"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "model.save('asl_model_84_54.h5')"
    ],
    "metadata": {
      "id": "KD44zAOOIL_7"
    },
    "execution_count": 19,
    "outputs": []
   }
  ]
}
```

TESTING THE MODEL

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

model = load_model('/content/Real_time.h5')

img = image.load_img('/content/Dataset/test_set/H/107.png',target_size = (100,100))
img
```



```
from skimage.transform import resize
def detect(frame):
   img=image.img_to_array(frame)
   img = resize(img,(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])
```

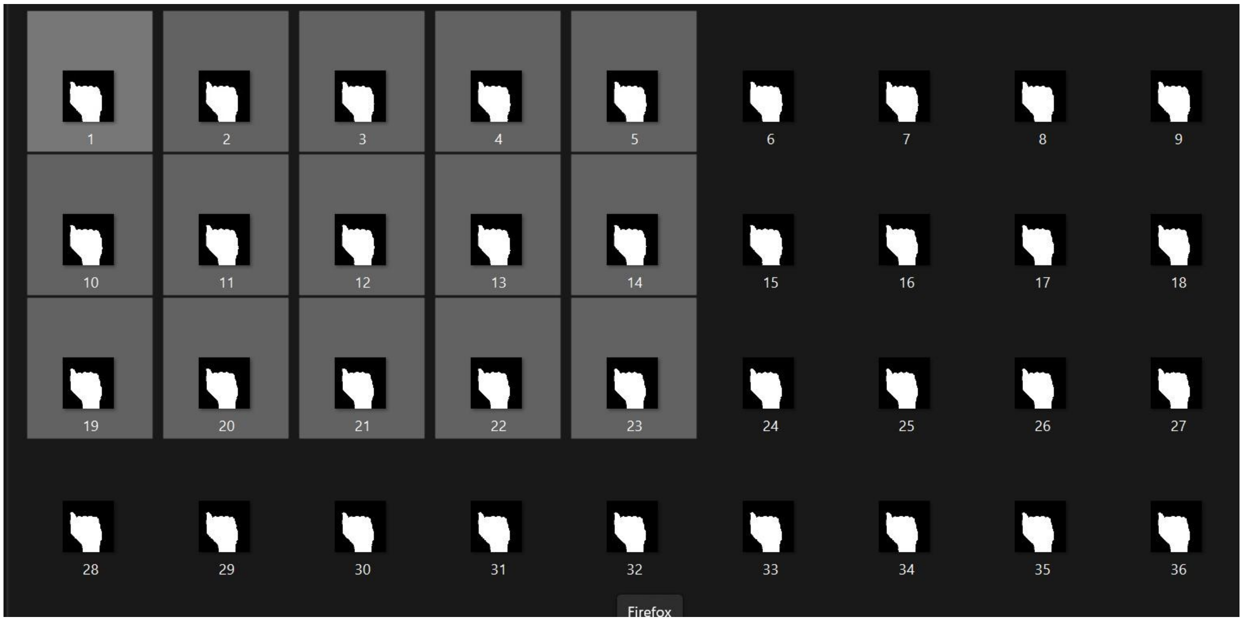
```
img=image.load_img("/content/Dataset/test_set/H/107.png")
detect(img)
1/1 [==============================] - 0s 28ms/step
THE PREDICTED LETTER IS H

img = image.load_img('/content/Dataset/test_set/A/110.png')
pred=detect(img)
1/1 [==============================] - 0s 26ms/step
THE PREDICTED LETTER IS A

img=image.load_img('/content/Dataset/test_set/E/111.png')
detect(img)
1/1 [==============================] - 0s 30ms/step
THE PREDICTED LETTER IS E

img=image.load_img('/content/Dataset/test_set/E/111.png')
detect(img)
1/1 [==============================] - 0s 30ms/step
THE PREDICTED LETTER IS E

img=image.load_img('/content/Dataset/test_set/E/111.png')
detect(img)
1/1 [==============================] - 0s 30ms/step
THE PREDICTED LETTER IS B

img=image.load_img('/content/Dataset/test_set/E/111.png')
detect(img)
1/1 [==============================] - 0s 30ms/step
THE PREDICTED LETTER IS C

img=image.load_img('/content/Dataset/test_set/E/111.png')
detect(img)
1/1 [==============================] - 0s 30ms/step
THE PREDICTED LETTER IS D

img=image.load_img('/content/Dataset/test_set/E/111.png')
```

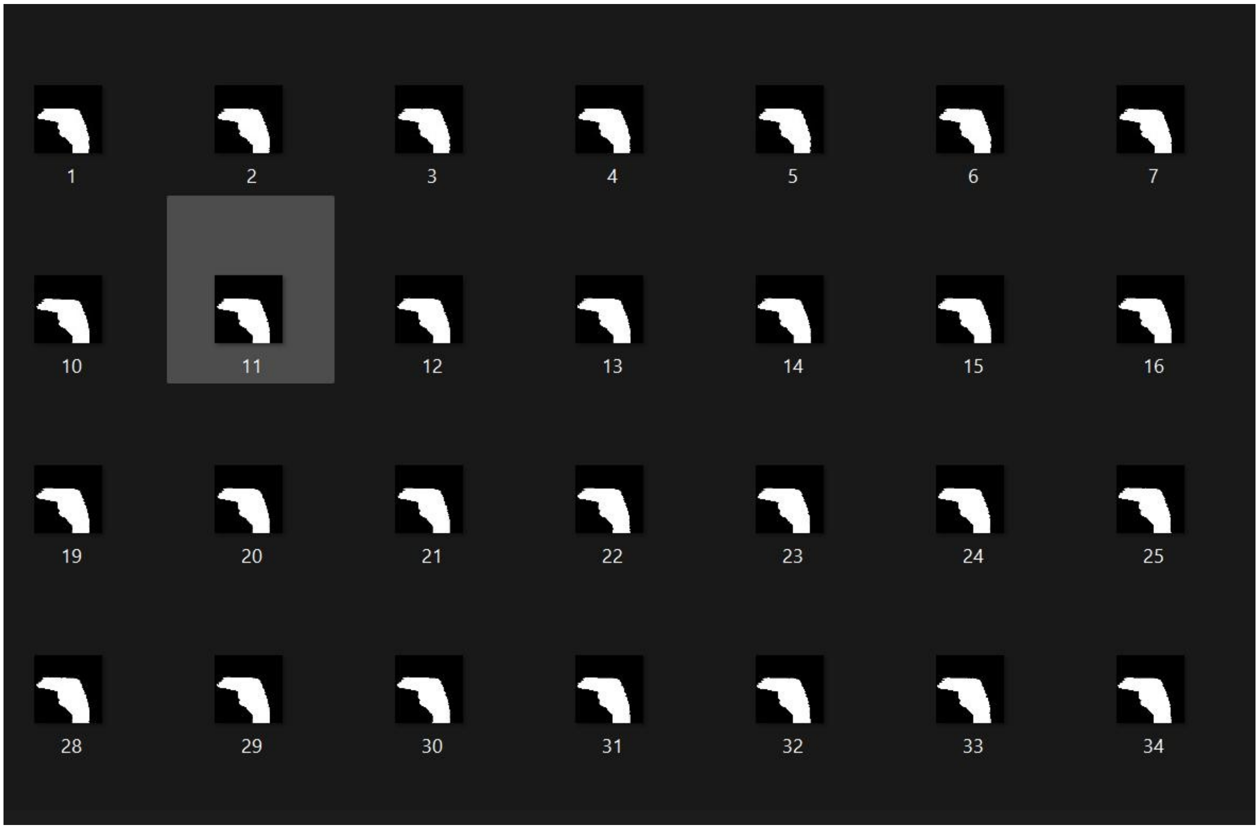
```
detect(img)
1/1 [==============================] - 0s 30ms/step
THE PREDICTED LETTER IS F

img=image.load_img('/content/Dataset/test_set/E/111.png')
detect(img)
1/1 [==============================] - 0s 30ms/step
THE PREDICTED LETTER IS G

img=image.load_img('/content/Dataset/test_set/E/111.png')
detect(img)
1/1 [==============================] - 0s 30ms/step
THE PREDICTED LETTER IS H
```

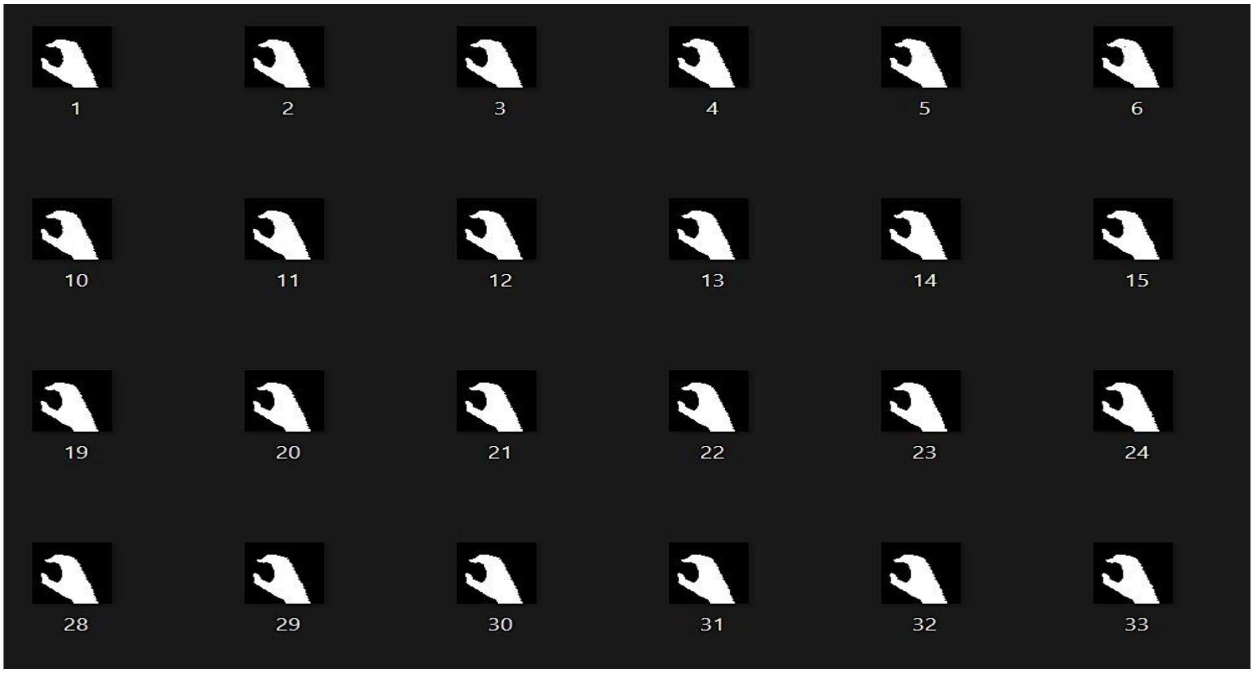
final spyder code deployment

```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "c497eb4d",
   "metadata": {},
   "outputs": [],
   "source": [
    "import cv2 #mporting opencv Library this i to open camera and take the video\n",
    "import numpy as np # to convert image to array and expand dimensions\n",
    "from tensorflow.keras.models import load_model # to Load the saved model\n",
    "from tensorflow.keras.preprocessing import image # to preproccess the image\n",
    "model = load_model(\"dataset.h5\") # we are loading the saved moodek\n",
    "video = cv2.VideoCapture(0) # two parameters 1, bool 0 or 1, frame\n",
    "index = [\"A\",\"B\",\"C\",\"D\",\"E\",\"F\",\"G\",\"H\",\"I\"]\n",
    "index=['A','B','C','D','E','F','G','H','I']\n",
    "#from playsound import playsound\n",
    "while(1):\n",
    "    success,frame = video.read()\n",
    "    cv2.imwrite(\"image.jpg\",frame)\n",
    "    img = image.load_img(\"image.jpg\",target_size = (64,64))\n",
    "    x = image.img_to_array(img)\n",
    "    x = np.expand_dims (x,axis = 0)\n",
    "    pred = np.argmax(model.predict(x),axis=1)\n",
    "    p = index [pred[0]]\n",
    "    print(\"predicted letter is: \"+ str(p))\n",
    "    #playSound(\"letter\"+str(str(index [p])+\"is detected\"))\n",
    "    cv2.putText (frame, \"predicted letter is \"+str(p), (100, 100), cv2.
FONT_HERSHEY_SIMPLEX, 1,(0,0,0), 4)\n",
    "    cv2.imshow(\"showcasewindow\", frame)\n",
    "    \n",
    "    if cv2.waitkey(1) & 0xFF == ord('a'):\n",
    "        break\n",
    "video.release()\n",
    "cv2.destroyAllwindows()"
   ]
  }
 ],
 "metadata":
  { "kernelspec":
  {
   "display_name": "Python 3.10.2 64-bit",
   "language": "python",
   "name": "python3"
  },
  "language_info":
   { "codemirror_mode":
   {"name": "ipython",
   "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.10.2"
  },
  "vscode": {
   "interpreter": {
    "hash": "695a388a4c2e020e22268ccf38be8173707e8975f5964aead99e22ad28ea09a9"
   }
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```

7.3 DATABASE SCHEMA




1
2
3
4
5
6
7
10
11
12
13
14
15
16
19
20
21
22
23
24
25
28
29
30
31
32
33
34

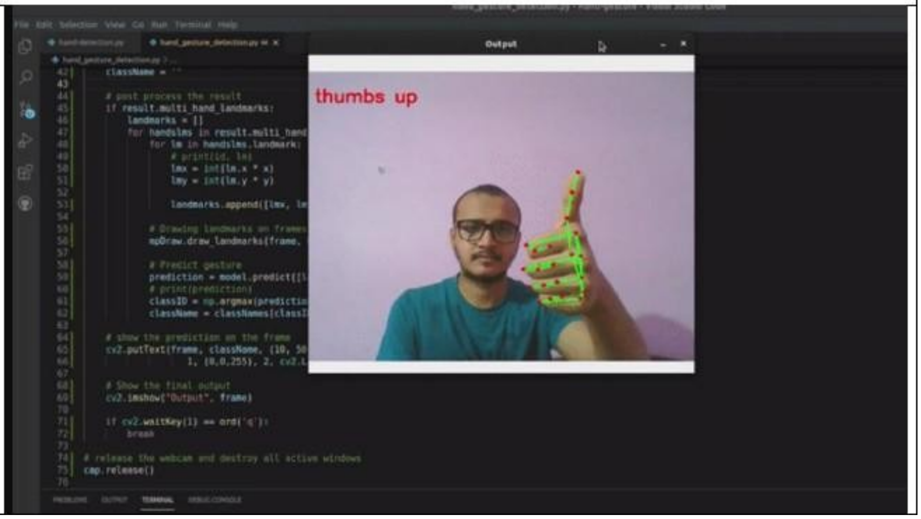
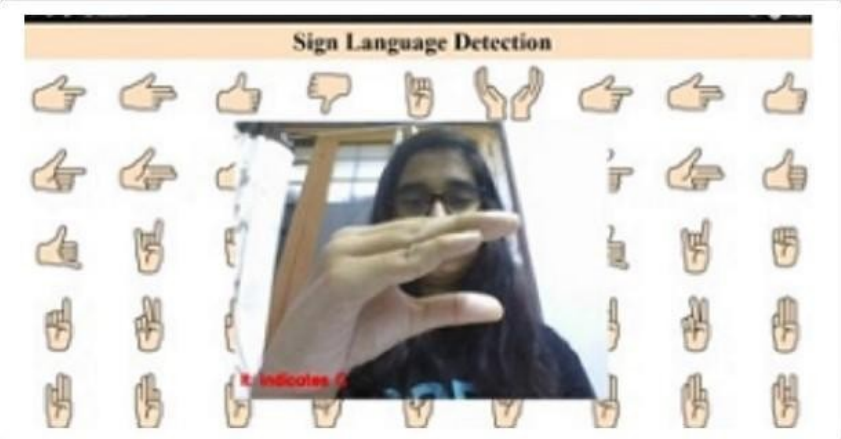

1
2
3
4
5
6
10
11
12
13
14
15
19
20
21
22
23
24
28
29
30
31
32
33

8. Testing

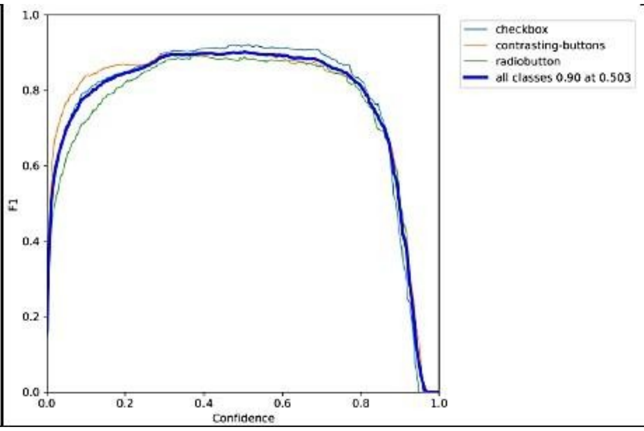
8.1 Testcase

Model Performance Testing:

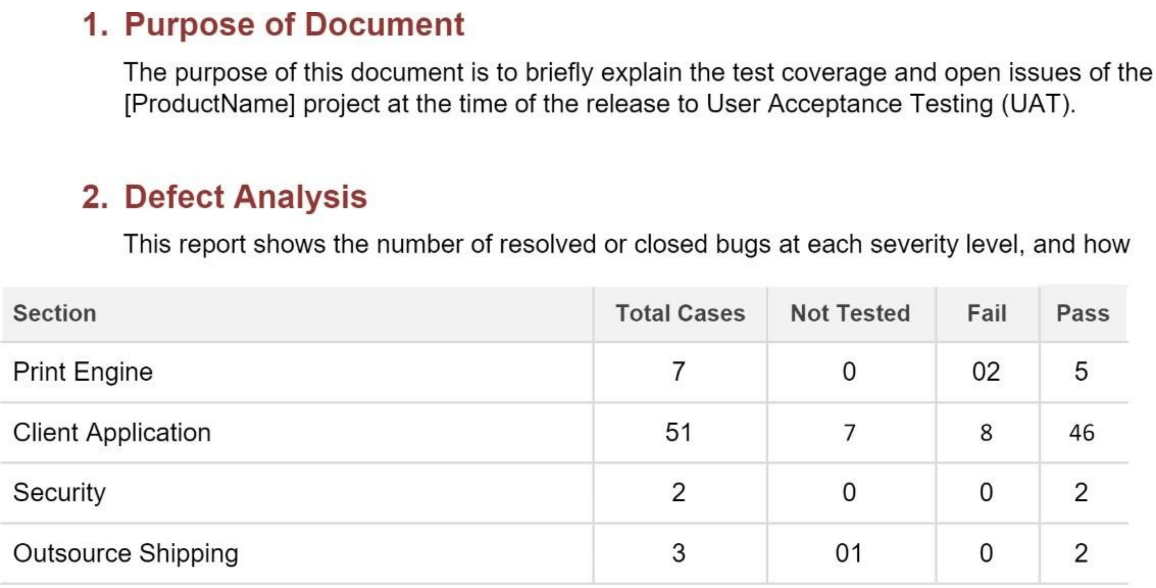
Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	
2.	Accuracy	Training Accuracy - 85 Validation Accuracy - 80	

```
        className = ''

        # post process the result
        if result.multi_hand_landmarks:
            landmarks = []
            for handslms in result.multi_hand
                for lm in handslms.landmark:
                    # print(id, lm)
                    lmx = int(lm.x * x)
                    lmy = int(lm.y * y)

                    landmarks.append([lmx, lm

                # Drawing landmarks on frames
                mpDraw.draw_landmarks(frame,

                # Predict gesture
                prediction = model.predict([[l
                # print(prediction)
                classID = np.argmax(predictio
                className = classNames[classI

        # show the prediction on the frame
        cv2.putText(frame, className, (10, 50
                    1, (0,0,255), 2, cv2.L

        # Show the final output
        cv2.imshow("Output", frame)

        if cv2.waitKey(1) == ord('q'):
            break

# release the webcam and destroy all active windows
cap.release()
```

3.	Confidence Score (Only Yolo Projects)	Class Detected - 8 Confidence Score – 75	

8.2 User Acceptance Testing



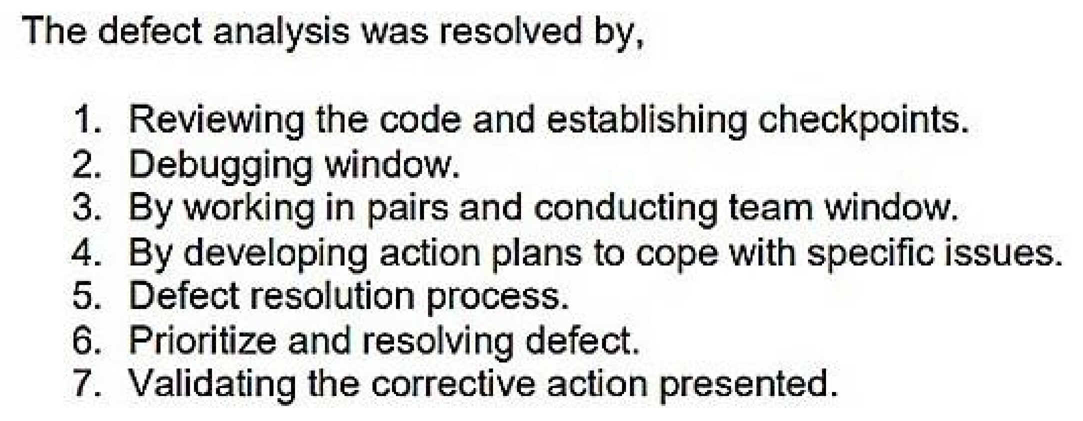
1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

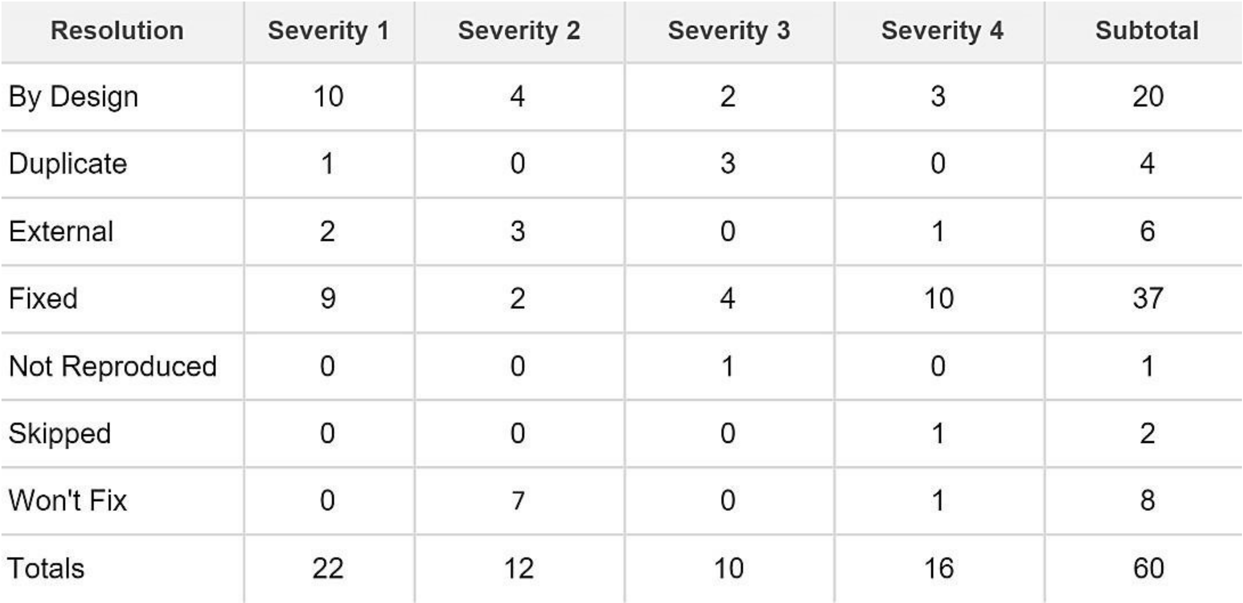
This report shows the number of resolved or closed bugs at each severity level, and how

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	02	5
Client Application	51	7	8	46
Security	2	0	0	2
Outsource Shipping	3	01	0	2



The defect analysis was resolved by,

1. Reviewing the code and establishing checkpoints.
2. Debugging window.
3. By working in pairs and conducting team window.
4. By developing action plans to cope with specific issues.
5. Defect resolution process.
6. Prioritize and resolving defect.
7. Validating the corrective action presented.



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	9	2	4	10	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	1	2
Won't Fix	0	7	0	1	8
Totals	22	12	10	16	60

9. RESULTS

9.1 Performance Metrics

Software quality is a measurement of something intangible, “how good” a software product really is. Some of the aspects of software quality taken are

a. Scalability

b. Speed

c. Stability

d. Reliability

e. Security

f. Maintainability and code quality

LOAD TEST

Scenario Name	Load Test – Ral time communication powered by AI for specially abled
Scenario Type	Load Test – Duration 1 hour
Scenario Objective	To Simulate the peak load and to monitor the performance of the Website
Steps	The online load will be maintained at steady state

Entry Criteria	All the monitors are in ready state
Exit Criteria	Response met the criteria and test completion report is agreed

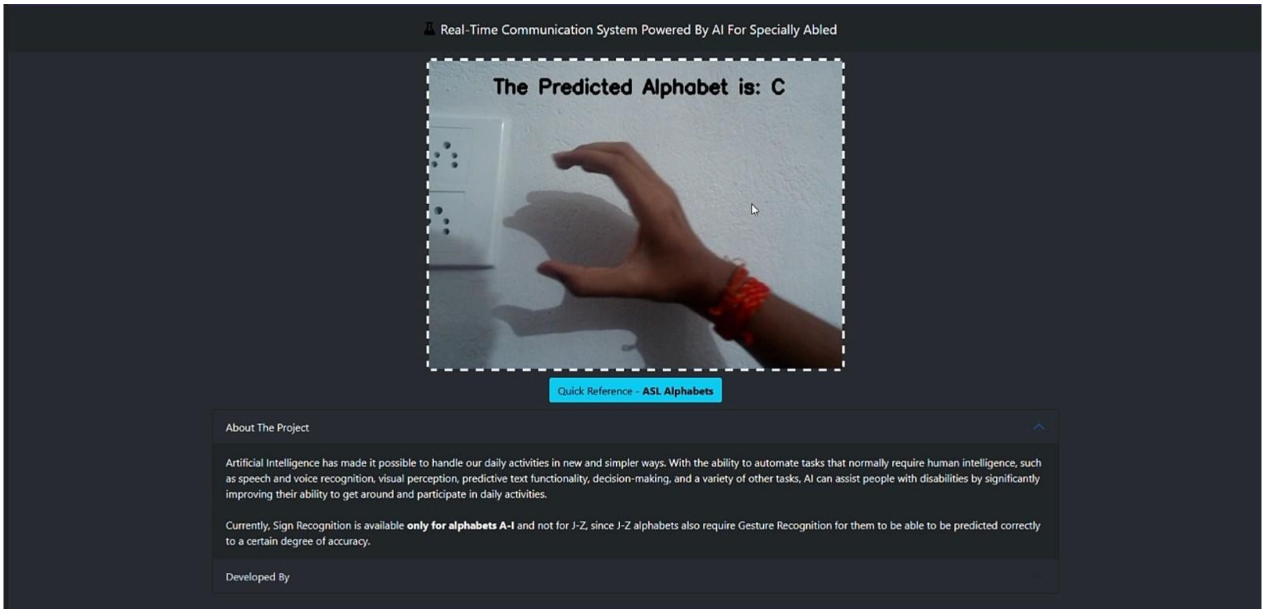
STRESS TEST

Scenario Name	Stress Test - Ral time communication powered by AI for specially abled
Scenario Type	Stress Test
Scenario Objective	Objective is to verify that the application can handle the projected growth and to discover the breaking point
Steps	Ramp up to 95% of peak volume and continuously increase load until breaking point
Entry Criteria	All the monitors are in place Test Data is set up Peak load test completed successfully
Exit Criteria	Test completion report is agreed upon as per expectation

ENDURANCE & SOAK TEST

Scenario Name	Soak Test –Ral time communication powered by AI for specially abled
Scenario Type	Endurance – Duration 8 hours
Scenario Objective	To discover memory issues and bottlenecks that might occur under daily usage of the application
Steps	Steady state is maintained for 8 hours with half of the peak load
Entry Criteria	All the monitors are in place Test Data is set up Peak load test completed successfully
Exit Criteria	Test completion report is agreed upon as per expectation

Execution results:

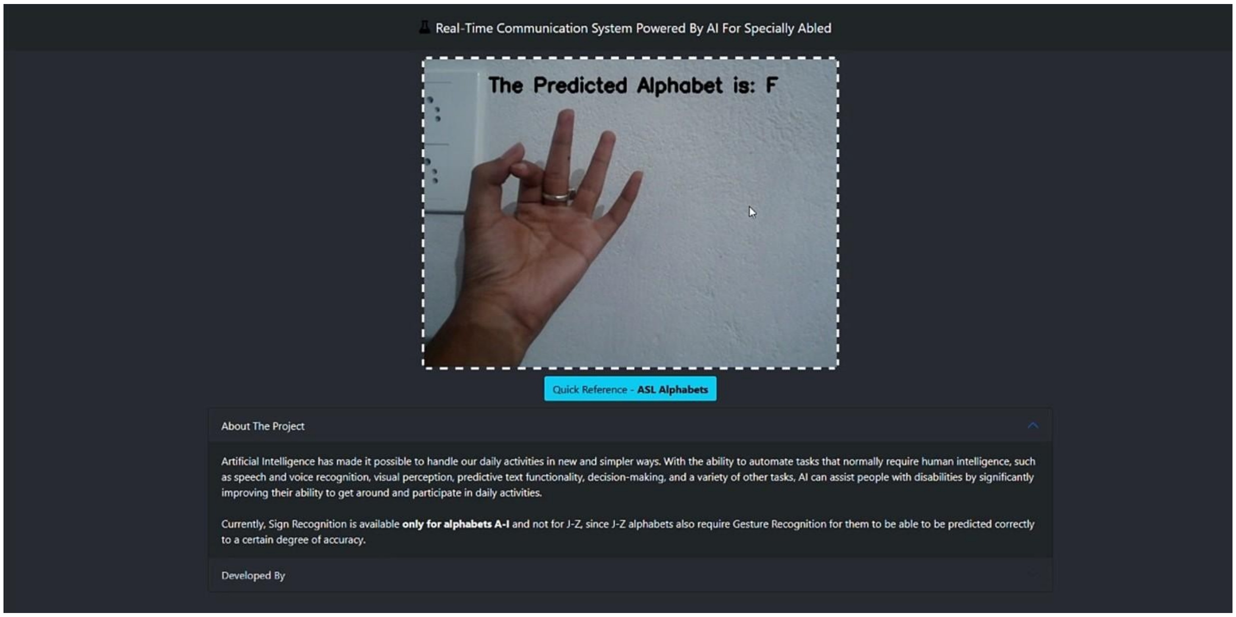


Real-Time Communication System Powered By AI For Specially Abled

The Predicted Alphabet is: C

Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly improving their ability to get around and participate in daily activities.

Currently, Sign Recognition is available **only for alphabets A-I** and not for J-Z, since J-Z alphabets also require Gesture Recognition for them to be able to be predicted correctly to a certain degree of accuracy.

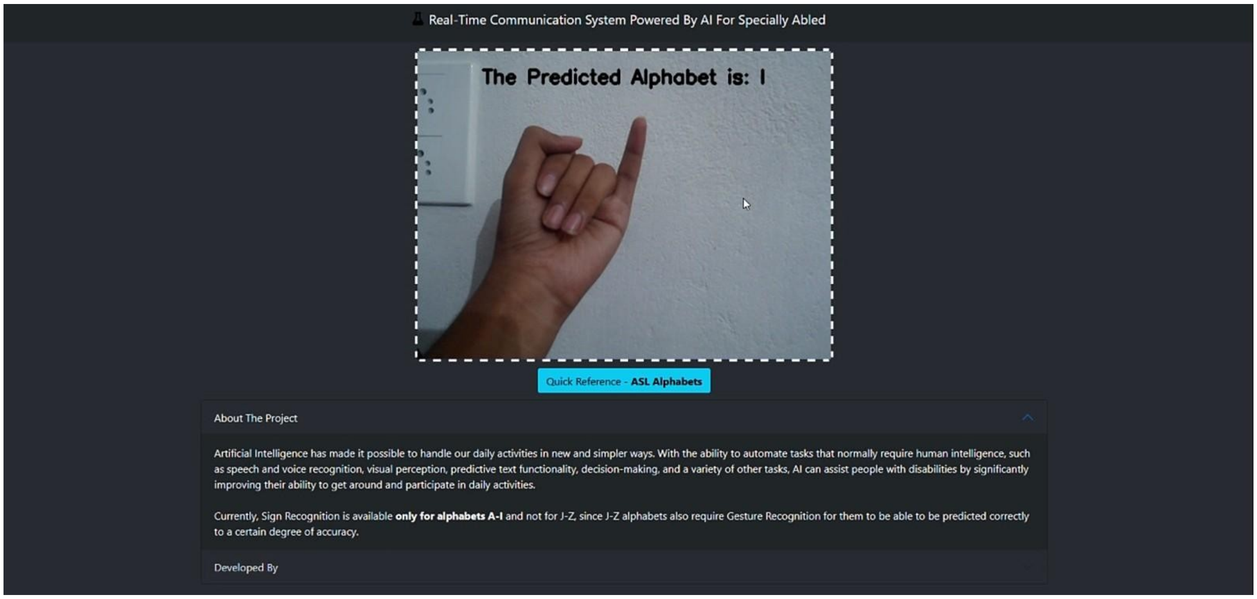


Real-Time Communication System Powered By AI For Specially Abled

The Predicted Alphabet is: F

Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly improving their ability to get around and participate in daily activities.

Currently, Sign Recognition is available **only for alphabets A-I** and not for J-Z, since J-Z alphabets also require Gesture Recognition for them to be able to be predicted correctly to a certain degree of accuracy.



Real-Time Communication System Powered By AI For Specially Abled

The Predicted Alphabet is: I

Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly improving their ability to get around and participate in daily activities.

Currently, Sign Recognition is available **only for alphabets A-I** and not for J-Z, since J-Z alphabets also require Gesture Recognition for them to be able to be predicted correctly to a certain degree of accuracy.

10. ADVANTAGES & DISADVANTAGES

Advantages:

1. It is possible to create a mobile application to bridge the communication gap between deaf and dumb persons and the general public.
2. As different sign language standards exist, their dataset can be added, and the user can choose which sign language to read.
4. Removing the communication barrier between the normal and abnormal people
5.Easy sign gesture converter to speech.

Disadvantages:

1. The current model only works from limited gestures.
2. In absence of sufficient gesture recognition cannot be identified as they require some kind of gesture input from the user.
3. As the quantity/quality of images in the dataset is low, the accuracy is not great, but that can easily be improved by change in dataset.

APPLICATIONS

1. It will contribute to the development of improved communication for the deafened. The majority of people are unable to communicate via sign language, which creates a barrier to communication.

2. As a result, others will be able to learn and comprehend sign language and communicate with the deaf and dumb via the web app.
3. According to scientific research, learning sign language improves cognitive abilities, attention span, and creativity.

11. CONCLUSION

Sign language is a useful tool for facilitating communication between deaf and hearing people. Because it allows for two-way communication, the system aims to bridge the communication gap between deaf people and the rest of society. The proposed methodology translates language into English alphabets that are understandable to humans.
This system sends hand gestures to the model, who recognises them and displays the equivalent Alphabet on the screen. Deaf-mute people can use their hands to perform sign language, which will then be converted into alphabets, thanks to this project.

12. FUTURE SCOPE

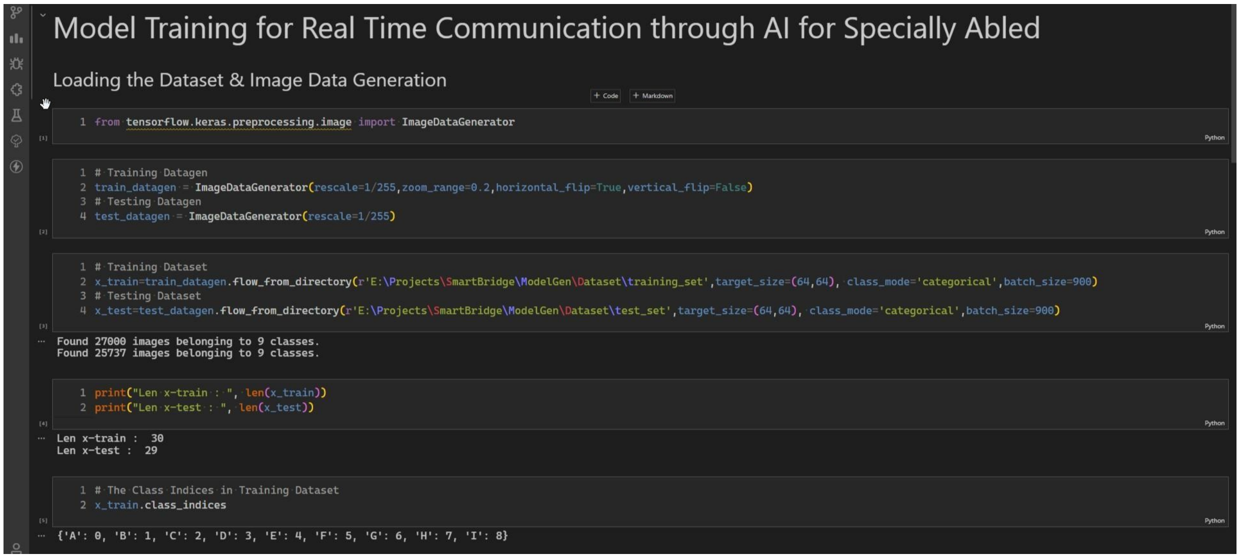
Having a technology that can translate hand sign language to its corresponding alphabet is a game changer in the field of communication and Ai for the specially abled people such as deaf and dumb. With introduction of gesture recognition, the web app can easily be expanded to recognize letters beyond 'I', digits and other symbols plus gesture recognition can also allow controlling of software/hardware interfaces.

We can develop the speech output to adaptive region of language.

13. APPENDIX

Source Code

Python code: Refer section 7 - Coding and Solutioning
Source Code for Model Training and Saving:



Model Training for Real Time Communication through AI for Specially Abled

Loading the Dataset & Image Data Generation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'E:\Projects\SmartBridge\ModelGen\Dataset\training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

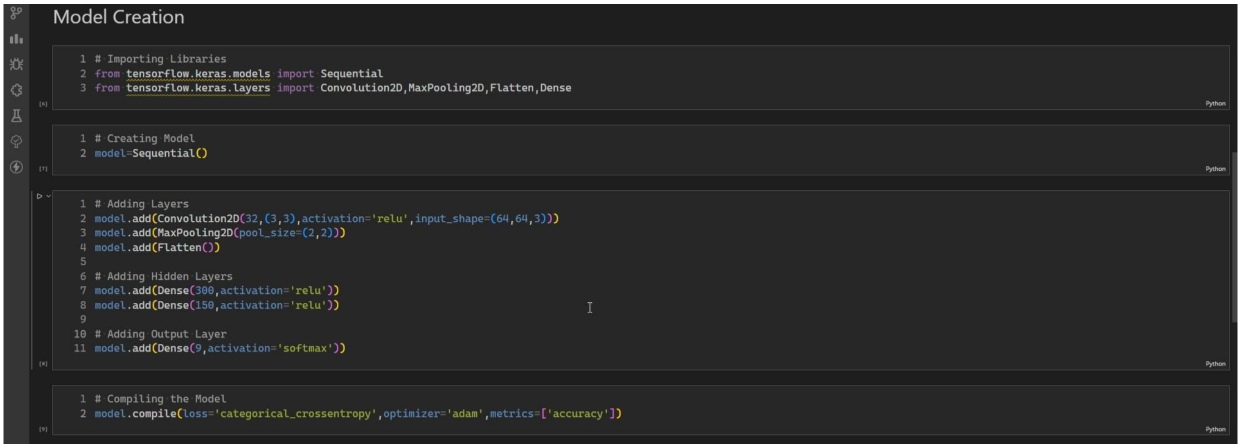
```
Found 27000 images belonging to 9 classes.
Found 25737 images belonging to 9 classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train :  30
Len x-test :  29
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```



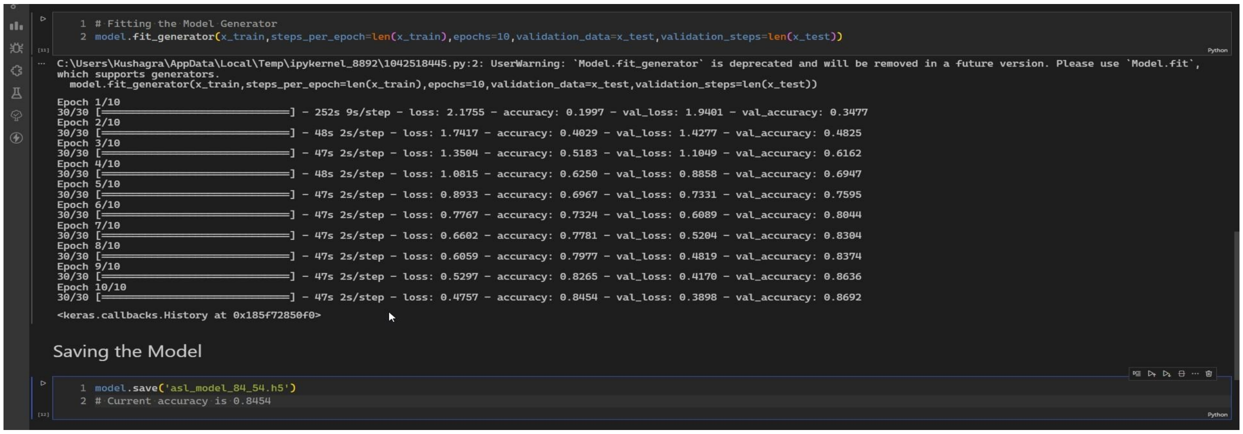
Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
# Creating Model
model=Sequential()
```

```
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

# Adding Hidden Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))

# Adding Output Layer
model.add(Dense(9,activation='softmax'))
```

```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```



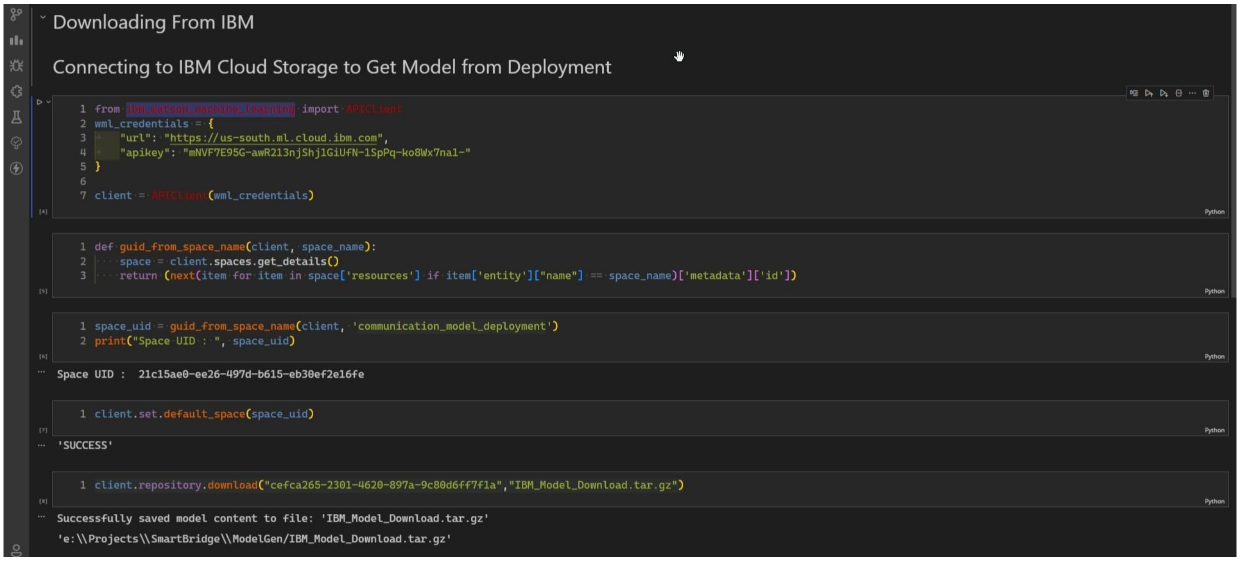
```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

```
C:\Users\Kushagra\AppData\Local\Temp\ipykernel_8892\1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`,
which supports generators.
  model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))

Epoch 1/10
30/30 [==============================] - 252s 9s/step - loss: 2.1755 - accuracy: 0.1997 - val_loss: 1.9401 - val_accuracy: 0.3477
Epoch 2/10
30/30 [==============================] - 48s 2s/step - loss: 1.7417 - accuracy: 0.4029 - val_loss: 1.4277 - val_accuracy: 0.4825
Epoch 3/10
30/30 [==============================] - 47s 2s/step - loss: 1.3504 - accuracy: 0.5183 - val_loss: 1.1049 - val_accuracy: 0.6162
Epoch 4/10
30/30 [==============================] - 48s 2s/step - loss: 1.0815 - accuracy: 0.6250 - val_loss: 0.8858 - val_accuracy: 0.6947
Epoch 5/10
30/30 [==============================] - 47s 2s/step - loss: 0.8933 - accuracy: 0.6967 - val_loss: 0.7331 - val_accuracy: 0.7595
Epoch 6/10
30/30 [==============================] - 47s 2s/step - loss: 0.7767 - accuracy: 0.7324 - val_loss: 0.6089 - val_accuracy: 0.8044
Epoch 7/10
30/30 [==============================] - 47s 2s/step - loss: 0.6602 - accuracy: 0.7781 - val_loss: 0.5204 - val_accuracy: 0.8304
Epoch 8/10
30/30 [==============================] - 47s 2s/step - loss: 0.6059 - accuracy: 0.7977 - val_loss: 0.4819 - val_accuracy: 0.8374
Epoch 9/10
30/30 [==============================] - 47s 2s/step - loss: 0.5297 - accuracy: 0.8265 - val_loss: 0.4170 - val_accuracy: 0.8636
Epoch 10/10
30/30 [==============================] - 47s 2s/step - loss: 0.4757 - accuracy: 0.8454 - val_loss: 0.3898 - val_accuracy: 0.8692

<keras.callbacks.History at 0x185f72850f0>
```

Saving the Model

```
model.save('asl_model_84_54.h5')
# Current accuracy is 0.8454
```

IBM Model Training & Download Code:



Downloading From IBM

Connecting to IBM Cloud Storage to Get Model from Deployment

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "mNVF7E95G-awR213njShj1GiUfN-1SpPq-ko8Wx7na1-"
}

client = APIClient(wml_credentials)
```

```
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return (next(item for item in space['resources'] if item['entity']["name"] == space_name)['metadata']['id'])
```

```
space_uid = guid_from_space_name(client, 'communication_model_deployment')
print("Space UID : ", space_uid)
```

```
Space UID :  21c15ae0-ee26-497d-b615-eb30ef2e16fe
```

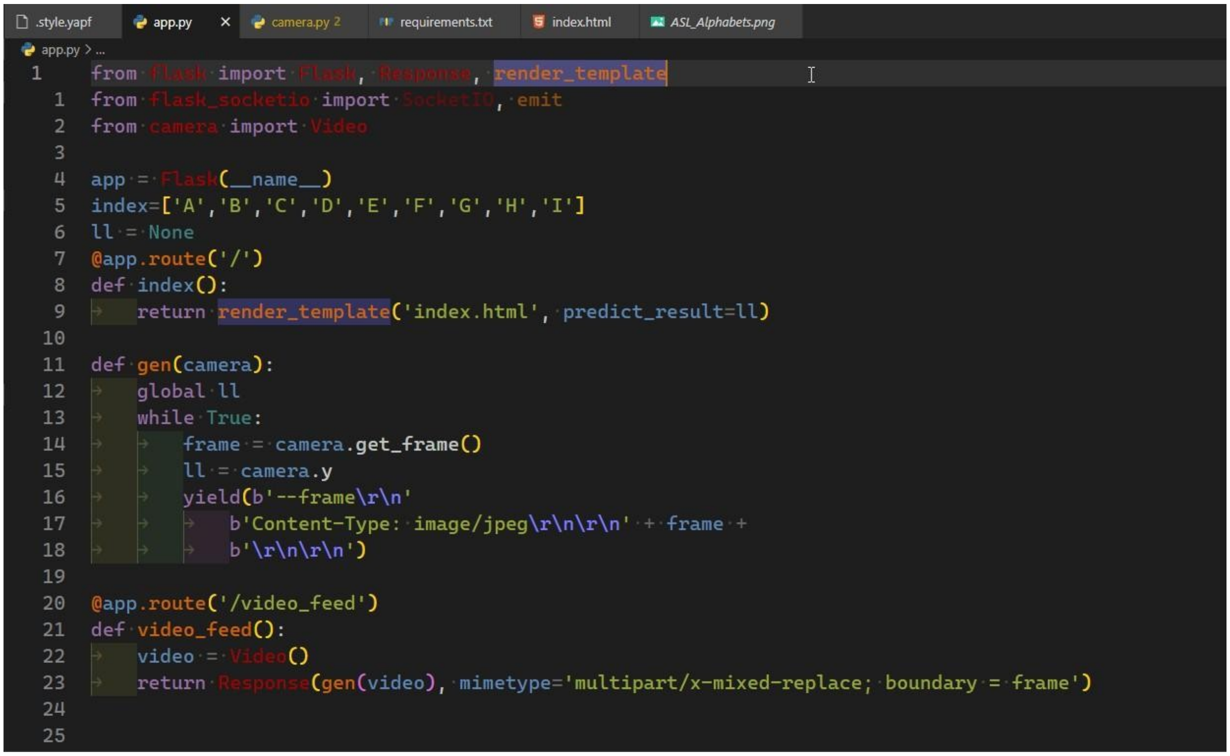
```
client.set.default_space(space_uid)
```

```
'SUCCESS'
```

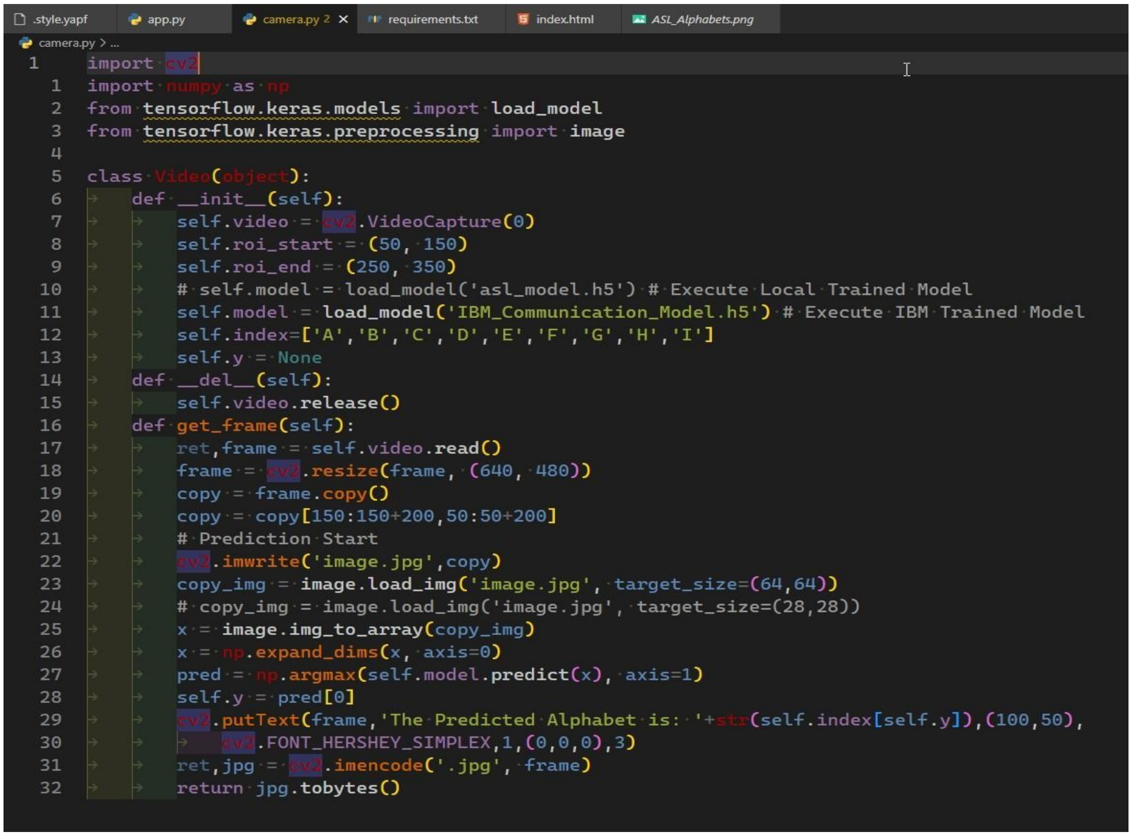
```
client.repository.download("cefca265-2301-4620-897a-9c80d6ff7f1a","IBM_Model_Download.tar.gz")
```

```
Successfully saved model content to file: 'IBM_Model_Download.tar.gz'
'e:\\Projects\\SmartBridge\\ModelGen/IBM_Model_Download.tar.gz'
```

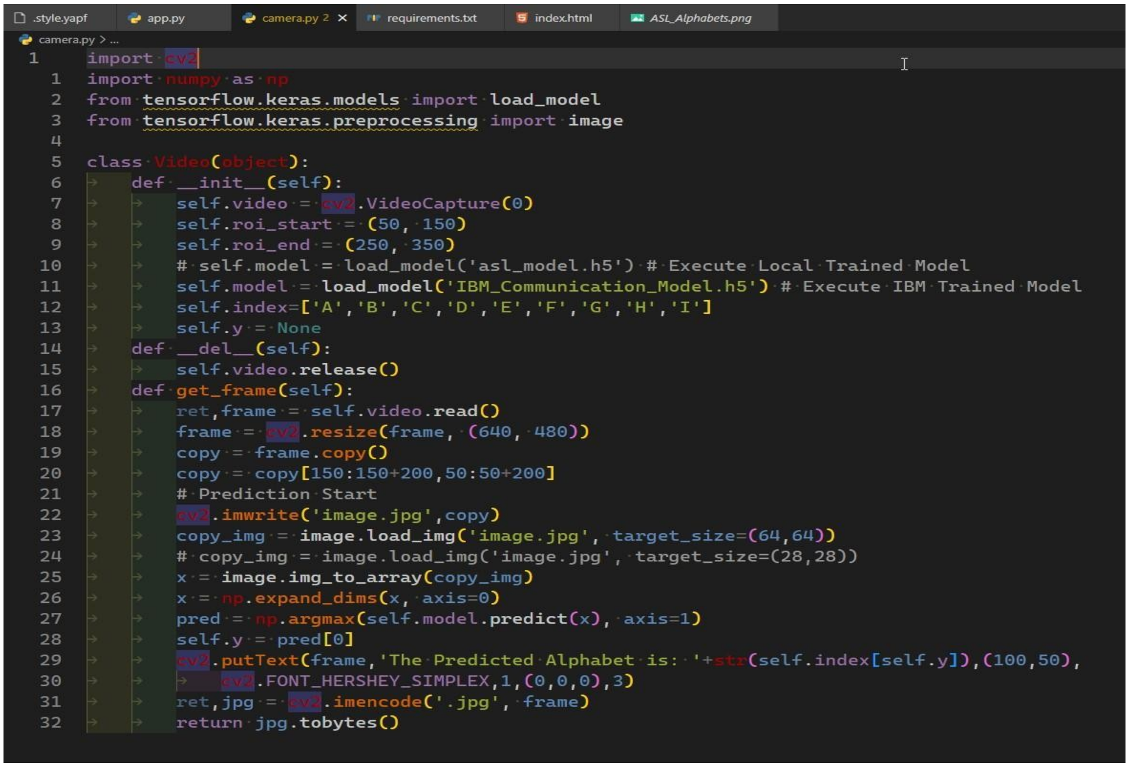
Web app code:



```
from flask import Flask, Response, render_template
from flask_socketio import SocketIO, emit
from camera import Video

app = Flask(__name__)
index=['A','B','C','D','E','F','G','H','I']
ll = None
@app.route('/')
def index():
    return render_template('index.html', predict_result=ll)

def gen(camera):
    global ll
    while True:
        frame = camera.get_frame()
        ll = camera.y
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +
            b'\r\n\r\n')

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



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

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        # self.model = load_model('asl_model.h5') # Execute Local Trained Model
        self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        # copy_img = image.load_img('image.jpg', target_size=(28,28))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),
            cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```


.style.yapf app.py camera.py 2 requirements.txt index.html ASL_Alphabets.png camera.py > ...

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

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        # self.model = load_model('asl_model.h5') # Execute Local Trained Model
        self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        # copy_img = image.load_img('image.jpg', target_size=(28,28))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),
            cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-18690-1659688505

DEMO VIDEO LINK :x

https://drive.google.com/file/d/1M74ztEPVd8gLUm1cDTRySTcRSFGWlBtq/view?usp=drivesdk