REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

TEAM ID - PNT2022TMID08592

S.No.	TEAM MEMBERS	REGISTER NUMBER
1	PRAVEEN KUMAR S (TL)	727619BCS031
2	SAKTHI KAVIN K	727619BCS013
3	LOGESH KUMAR S	727619BCS035
4	SHIVAPRIYAN K	727619BCS049

Project Report Format

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Database Schema (if Applicable)
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    - Source Code
    - GitHub & Project Demo Link

CHAPTER 1

INTRODUCTION

1.1 Project Overview

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. 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. 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.

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 image is given as output.

1.2 PURPOSE

The proposed idea 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.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Some of the existing solutions for solving this problem are:

Paper 1- Messaging and Video Calling Application for Specially Abled people using Hand Gesture Recognition

In the fifirst existing survey **R. R. Chhajed, K. P. Parmar, M. D. Pandya and N. G. Jaju, "Messaging and Video Calling Application for Specially Abled people using Hand Gesture Recognition," 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp. 1-4, doi: 10.1109/I2CT51068.2021.9417924.**

‘The existing paper proposes a system to overcome these barriers and allow everyone to interact with each other irrespective of their disabilities and facilitate everyone to communicate with each other through messaging and video calling irrespective of their disabilities. This paper proposes a vision-based application which can be used for the communication of such people using text and video calling. For better accuracy various object detection and image classification algorithms are implemented. The application uses Indian Sign Language as the dataset.

Paper 2 - IFSA: an integrated framework for developing IoT linked mobile applications for specially abled people

In the second existing survey **Kaur, S., Dhindsa, K.S. IFSA: an integrated framework for developing IoT linked mobile applications for specially abled people. *Wireless Netw* 28, 1375–1388 (2022)**

The existing paper proposed the Internet of Things has the potential to improve social interaction for visually challenged people. Hardware devices are constantly being equipped with various electronic sensors for collecting real-time data. However, specially-abled people need an integrated system to access the features of mobile applications and external hardware kits on one platform. Therefore, an integrated framework for the specially-abled is developed. IoT has the potential to improve social integration for people with visual defects. This research is an attempt to design a framework for developing a mobile application using IoT to provide secure and integrated services to the visually impaired people. The findings of the study revealed that the designed framework will help in developing various wireless embedded systems using mobile phones.

Paper 3- Sign Language Recognition System for Deaf People

In the third existing survey **Sharma, A., Pingale, S., Sabale, U., Patil, N., Dongre, S. (2023). Sign Language Recognition System for Deaf People. In: Garg, D., Kumar, N., Iqbal, R., Gupta, S. (eds) Innovations in Information and Communication Technologies. Algorithms for Intelligent Systems. Springer, Singapore.**

A sign language recognition system is a way to communicate with deaf–mute people. A large number of deaf and mute people are present across the world, and sometimes, it becomes difficult for normal people to communicate with them since not everyone can understand sign language. To establish effective communication between normal and specially abled people, there is a need to encourage the use of a sign language recognition system. In this language, people communicate through various hand gestures with each other. The purpose of language is to bridge the gap between the deaf–mute communities and the speaking folks. This research proposes an optimal recognition system whose major objective is to accomplish the translations of static sign language alphabets, numbers and words of American Sign Language into human and machine understandable English language. In the proposed model, in the first

phase, the preprocessing functionality of input gestures takes place. In the next phase, various region properties of preprocessed gestures will be computed by the system. In the final phase, based on the propertiescalculated before, the translation of sign to text is to be carried out and the same works in the opposite manner as well for speech to sign conversion.

Paper 4- A Robust Business Specific Real-Time Sign Language Translator

In the fourth existing survey **Waiz KhanStudent, Department of Computer Engineering, Khaja Bandanawaz College of Engineering, Kalaburagi, Karnataka, India**

Communication is a great way of expressing yourself but not everyone is capable of communication. No, not everyone chooses to be silent, While some are born deaf and mute, others become one later in life due to certain conditions. The word “dumb” is quite offensive so I will address to someone who cannot speak as mute person throughout this paper. Returning to the topic this paper will enable a more effective way to fill the communication gap between deaf and mute person and normal person especially in places with crowd where special-abled person becomes nervous and anxious to make the business owner understand of his needs easily and are unable to express themselves. In order to remove this barrier and enable the effective communication between the special-abled buyer and the business owner, I have created our own data set of hand gestures and trained using Google Teachable Machine for common sentences that are exchanged between the buyer and seller**.**

Paper 5- A Deep Learning Framework for Real-Time Indian Sign Language Gesture Recognition and Translation to Text and Audio

In the fifth existing survey **Deshpande, A.M., Inamdar, G., Kankaria, R., Katage,S. (2023). A Deep Learning Framework for Real-Time Indian Sign Language Gesture Recognition and Translation to Text and Audio. In: Pati, B., Panigrahi, C.R., Mohapatra, P., Li, KC. (eds) Proceedings of the 6th International Conference on Advance Computing and**

Intelligent Engineering. Lecture Notes in Networks and Systems, vol 428. Springer, Singapore.

Indian Sign Language (ISL) is used in the deaf community all over India.Development of the ISL recognition system is an active area to aid this community. In ISL, most of the signs are two-handed signs, and thus, it differs from another commonly used American Sign Language (ASL) and seems complex. In this paper, the design and implementation of a system to recognize ISL signs is reported. Building such a system can help specially abled person/people, by providing a medium to communicate withothers without human interpreters. The proposed system is built using a deep convolutional neural network (CNN), which performs both feature extraction and classification, preceded by an image preprocessing step. A real-time input (live signs captured from webcam) is given to this system, and the output is delivered in the form of text and audio

Paper 6- Sign Language Recognition Using Convolutional Neural Network

In the existing survey **Rakesh, S., Bharadhwaj, A., Sree Harsha, E. (2021). Sign Language Recognition Using Convolutional Neural Network. In: Raj, J.S., Iliyasu, A.M., Bestak, R., Baig, Z.A. (eds) Innovative Data Communication Technologies and Application. Lecture Notes on Data Engineering and Communications Technologies, vol 59. Springer, Singapore.**

In today's world, communication is very important. A language is needed to communicate. Most of the specially abled people, use a different language for communication called sign language. This language helps them to communicate with other people with their hand expressions. These expressions will be different from country to country. In this paper,

American sign language is used. This paper deals with helping specially abled people to communicate with people who don’t know sign language by using the approaches of computer vision and deep learning. Our paper uses convolutional neural network to solve this problem. The first part of our paper focuses on capturing different hand expressions in the form of video by the person and translating them to text using a convolutional neural network. The other part focuses on the reverse of it, showing GIF upon converting text. Integrating these two parts will help in two-way communication.

2.2 Problem Statement Definition

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 images are given as output.

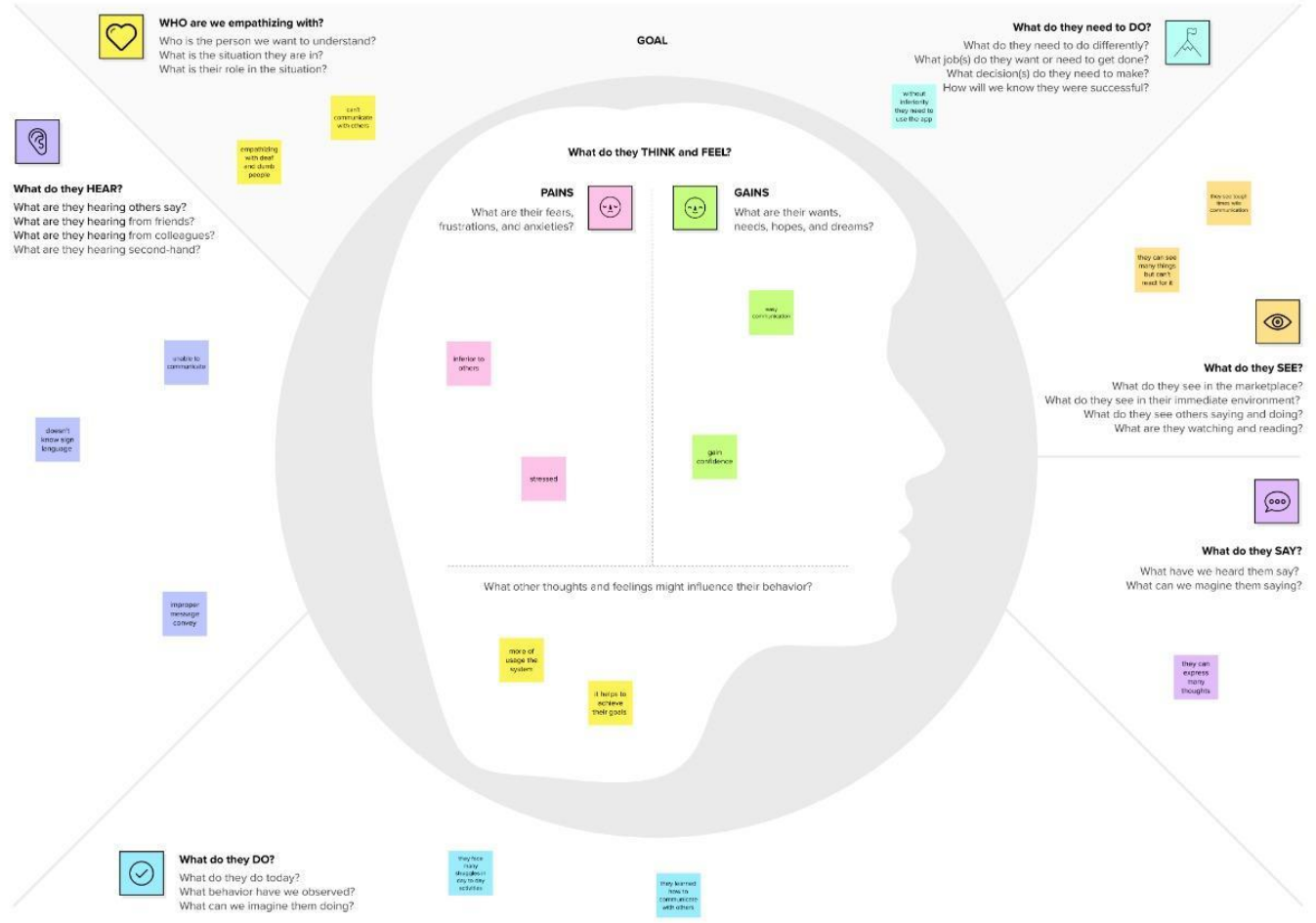
Example:

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Deaf Person	Communicate with others	I am unaware of the variety of solution available	I do not have correct guidance	Inferior to others
PS-2	Deaf and Dump Person	Convey my thoughts but peoples can't my language	I am unaware about the technology development	I am unable to know the current updates	Stressed

CHAPTER 3

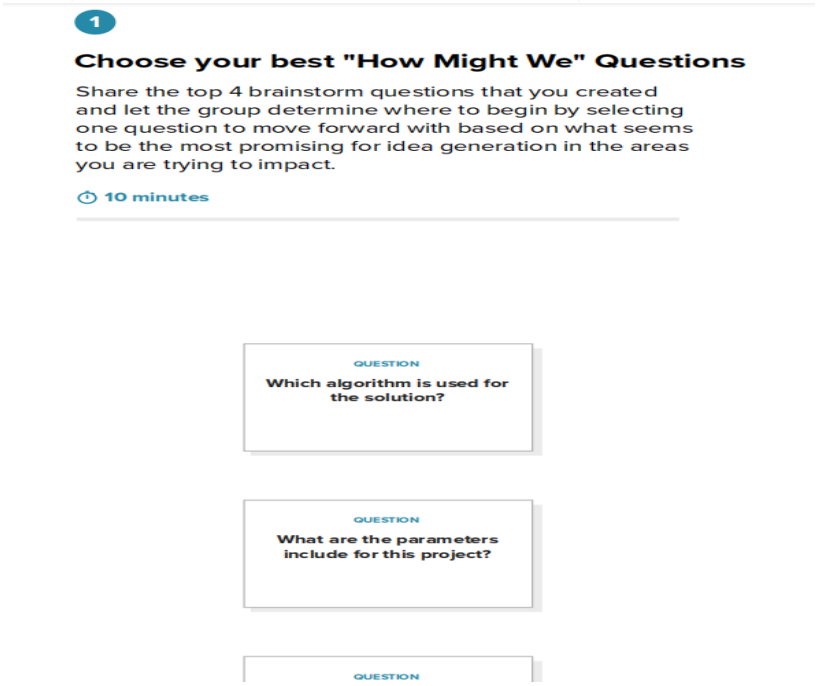
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

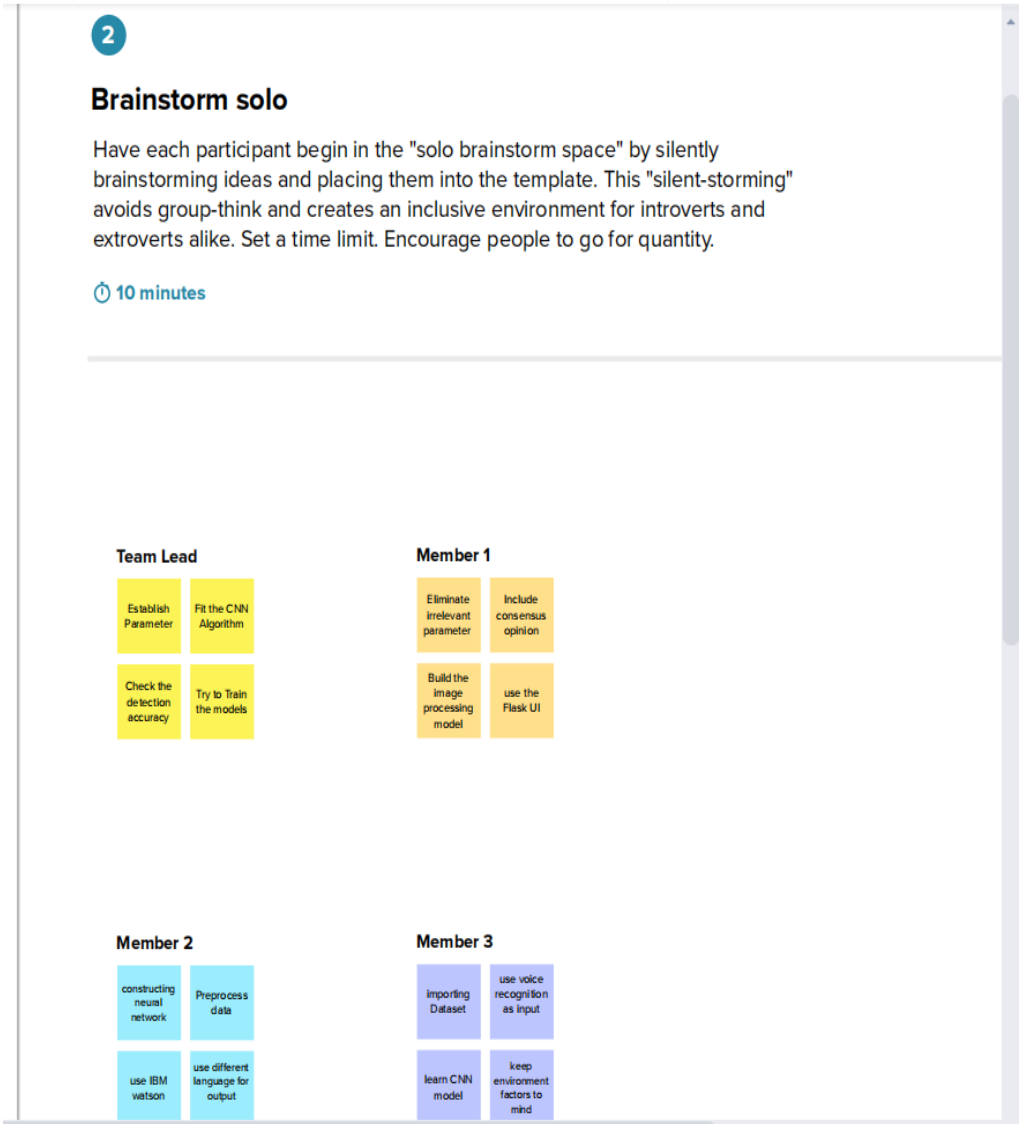


3.2 Ideation & Brainstorming

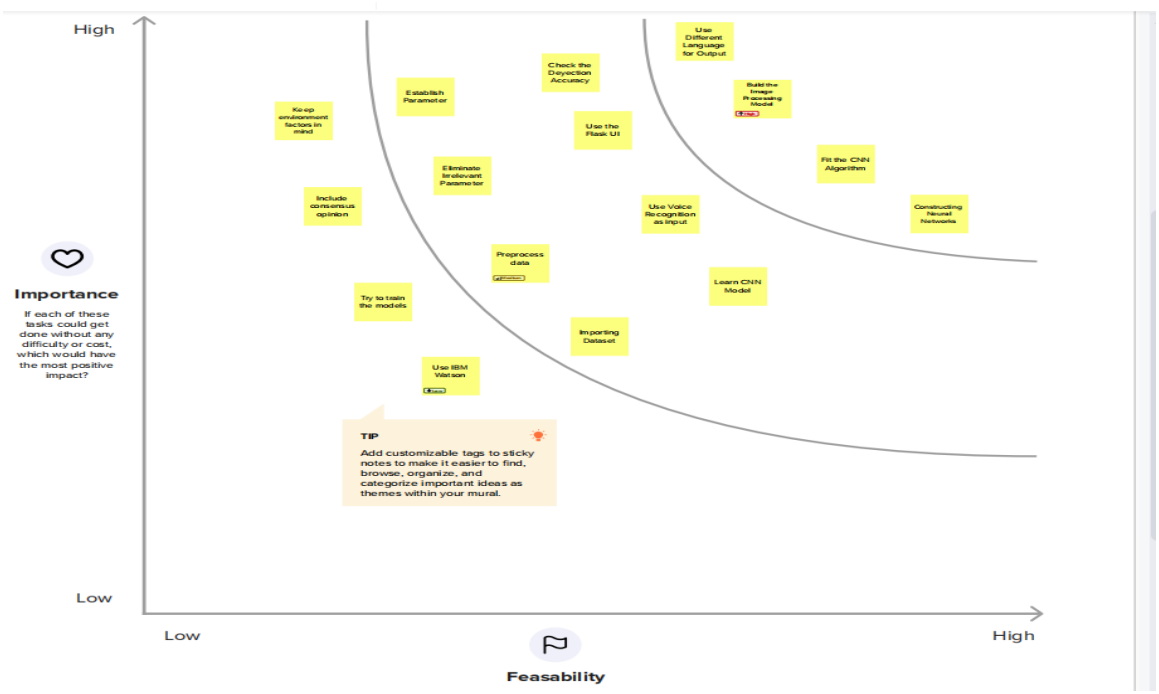
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

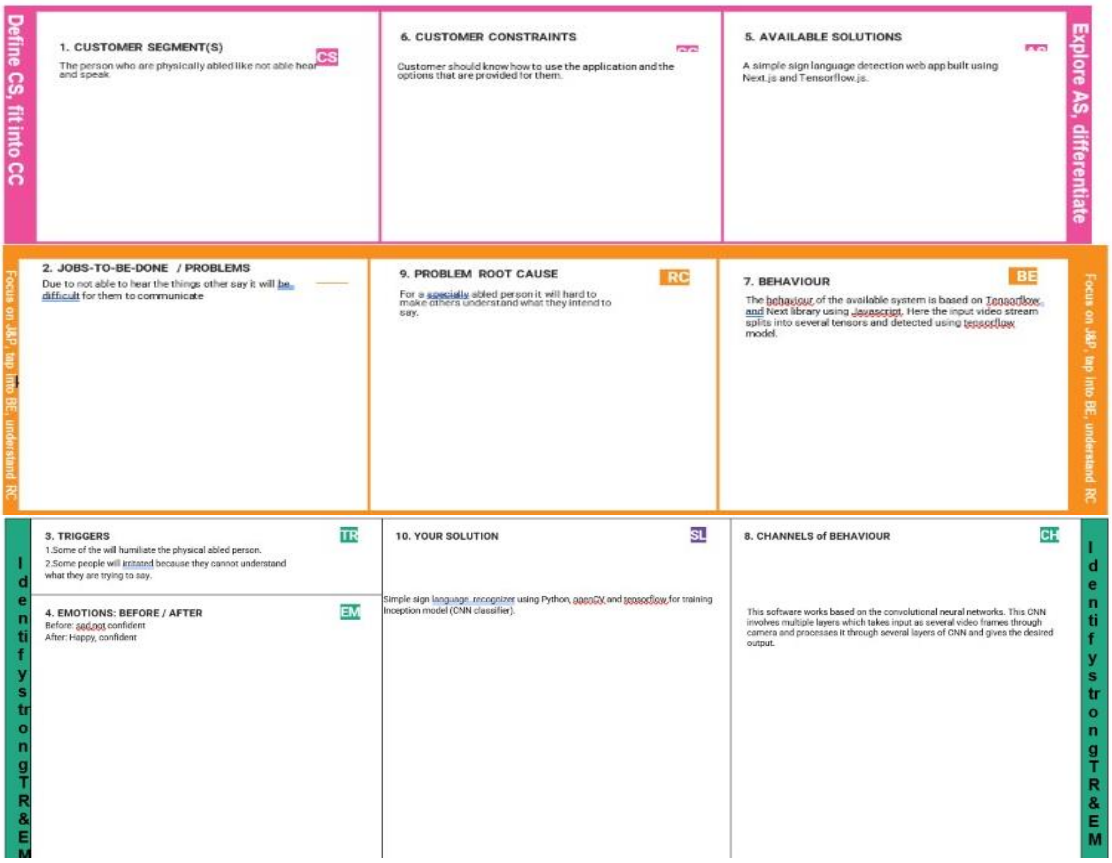


3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.
2.	Idea / Solution description	The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.
3.	Novelty / Uniqueness	This processes the image of the person who is using sign language and converts it into the voice by analyzing the sign used.
4.	Social Impact / Customer Satisfaction	Differently abled people feel free to communicate and it brings a huge difference compared to the past
5.	Business Model (Revenue Model)	There are many people in the world who are differently able,this application will become more popular among them and it will be installed by all and it will be used,and so it will produce more money

6.	Scalability of the Solution	Thus this would bring a new evolution in Real Time Communication System Powered by AI for Specially Able with less time and safe enough resources.

3.4 Problem Solution fit



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirements

Hardware Requirements:

Operating System	Windows, Mac, Linux
CPU (for training)	Multi Core Processors (i3 or above/equivalent)
GPU (for training)	NVIDIA AI Capable / Google's TPU
Web Cam	Integrated or External with Full HD Support

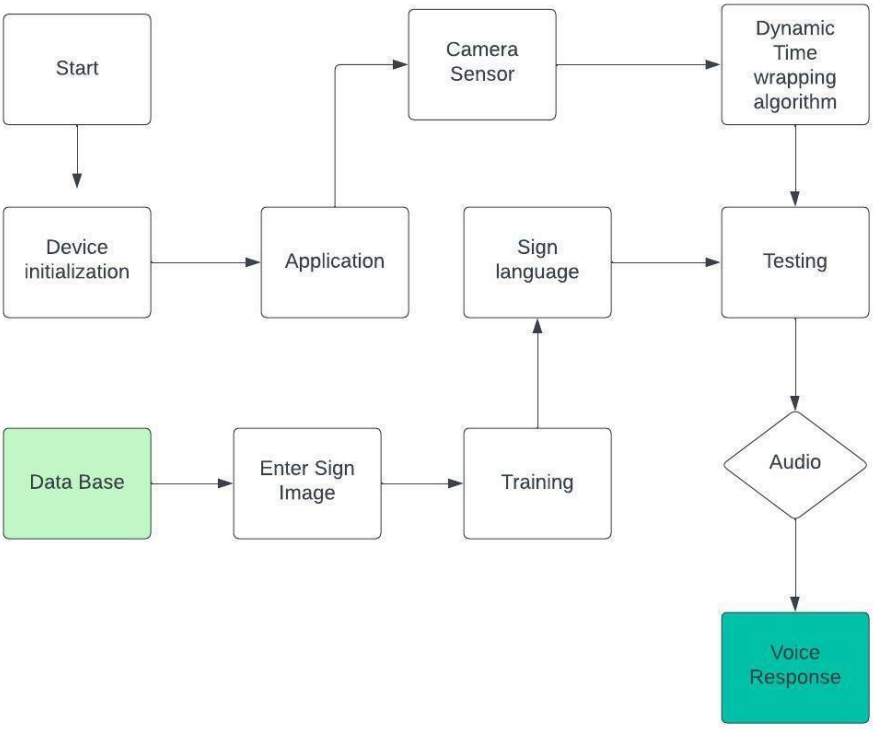
Software Requirements

Python	v3.9.0 or Above
Python Packages	flask, tensorflow, opencv-python, keras, numpy,pandas, virtualenv, pillow
Web Browser	Mozilla Firefox, Google Chrome or any modern web browser
IBM Cloud (for training)	Watson Studio - Model Training & Deployment as Machine Learning Instance

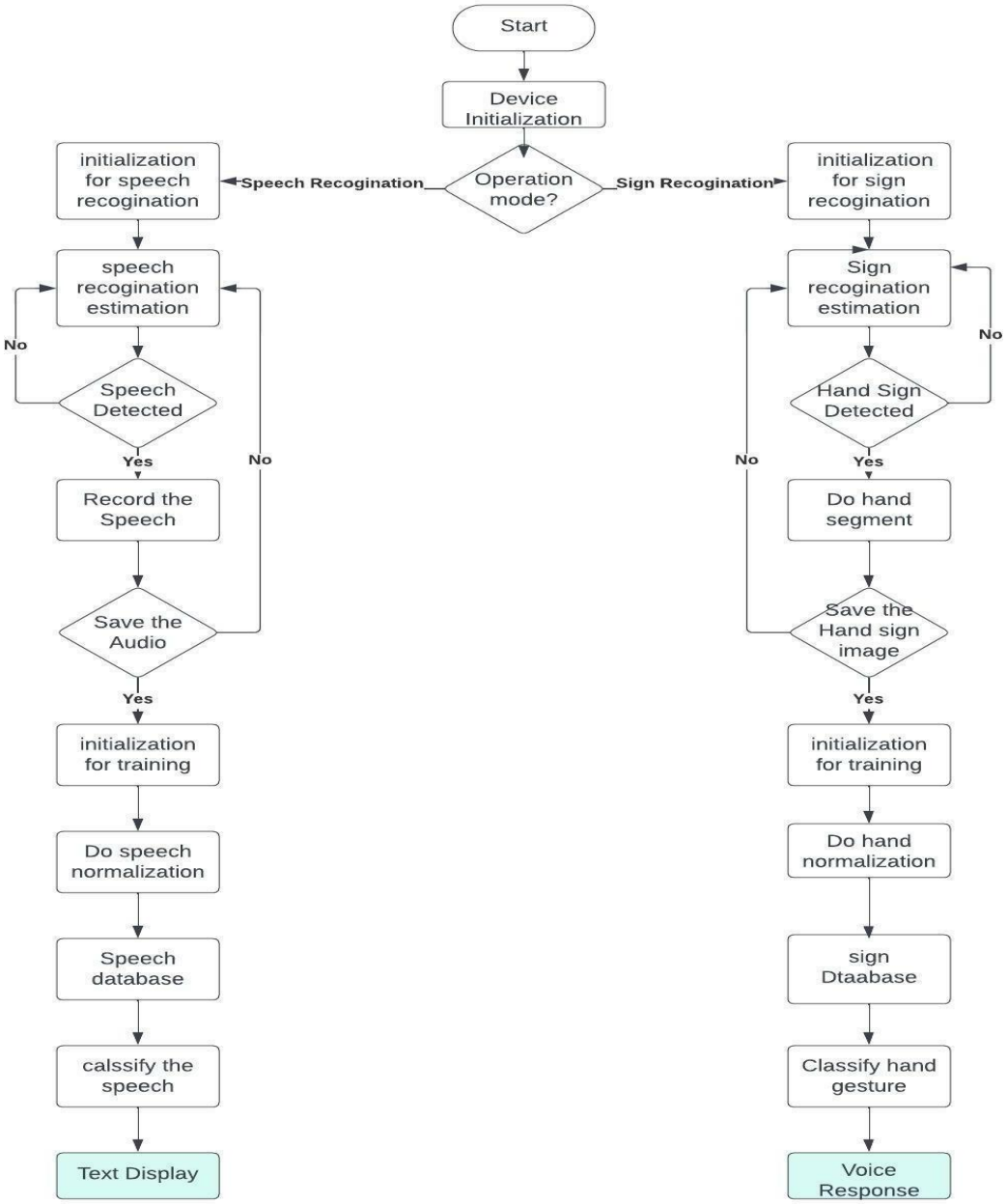
CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagrams



Data Flow Diagram



5.2 Solution & Technical Architecture

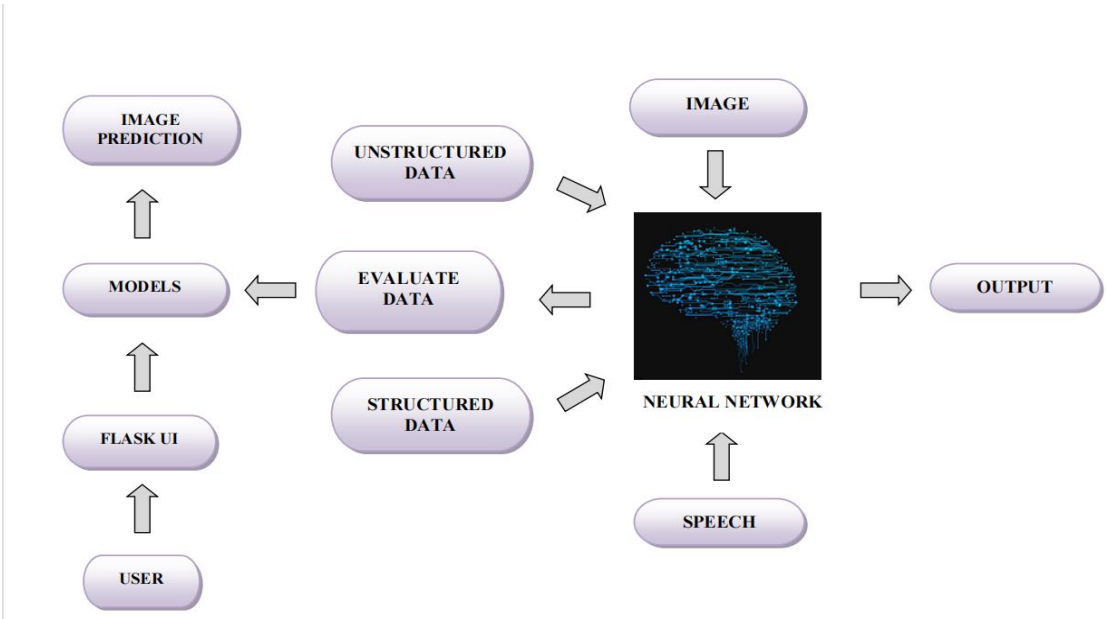
Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

.

Example - Solution Architecture Diagram:



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer *(Desktop user)*	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard	USN-3	As a user, I can select options in dashboard.	I can select options in dashboard.		
Customer *(Desktop user)*	**Main page**	USN-4	As a User, **I can enter the web page once clicked, which provides be the Guidelines** to use the app	I can **enter the web page** once clicked.	Medium	Sprint-1
Customer (Desktop user)	**Guidelines**	USN-5	As a User, **I can give a read through the guidelines to understand the functioning of the app.**	I can **give a read through the guidelines.**	Medium	Sprint-1
Customer *(Desktop user)*	**Convert Sign**	USN-6	As a User, **I can click the button Convert sign**, which directs me towards the Main screen	I can **click the button Convert sign and it direct me to main screen.**	Medium	Sprint-2
Customer *(Desktop user)*	**Camera (Hand movement detection)**	USN-7	As a User, **I can show my hand sign towards the camera which converts them into text manner.**	**I can show my hand sign** towards the camera accurately.	High	Sprint-2
Customer *(Desktop user)*	**Voice mode**	USN-8	Once the text is obtained, as a User **I can click on the voice mode which provides the text**	**I can click on the voice mode** which provides the text in	High	Sprint-3
Customer Care Executive	**Provide the necessary functionalities required to use the app.**	USN-9	As an Executive, I can **provide the Specifications of Camera required, and other factors** that are required for smooth functioning of the app.	**I can provide the Specifications** of camera required, and other factors	Low	Sprint-1
Customer Care Executive	**Check the performance of the app**	USN-10	As an Executive, I can **check the usage and queries obtained from the end users**.	I can **check the usage and queries** obtained from the end users.	Medium	Sprint-1
Administrator	**Receive queries based on usage**	USN-11	As an Admin, I can **take the queries from the customer care and perform the testing phase again, loading the other signs in the dataset**, in order to make the customers to use the app effectively.	I can **take the queries from the customer care and perform necessary phases again.**	High	Sprint-3

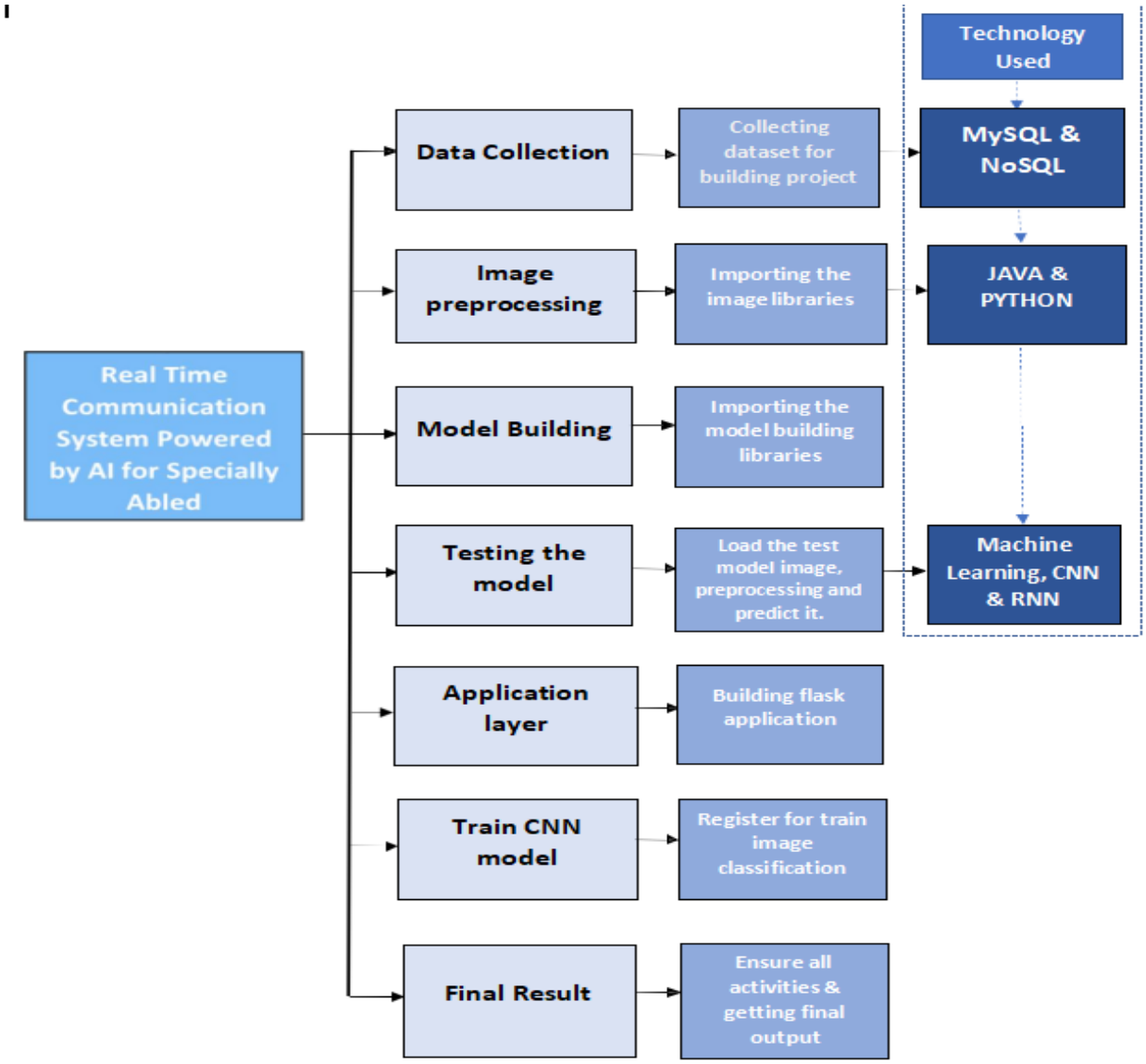
CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning And Estimation

Milestone	Function (Epic)	Milestone Story Number	Story / Task
Milestone 1	Data collection	M1	we're collecting dataset for building our project and creating two folders, one for training and another one for testing.
Milestone 2	Image preprocessing	M2	Importing image data generator librariesand applying image data generator functionality to train the test set.
Milestone 3	Model building	M3	Importing the model building libraries, Initializing the model, Adding Convolution layers, Adding the Pooling layers, Addingthe Flatten layers, Adding Dense layers, Compiling the model Fit and Save the model.
Milestone 4	Testing the model	M4	Import the packages first. Then we save the model and Load the test image, preprocess it and predict it.
Milestone 5	Application layer	M5	Build the flask application and the HTML pages.
Milestone 6	Train CNN model	M6	Register for IBM Cloud and train Image Classification Model.
Milestone 7	Final result	M7	To ensure all the activities and resultingthe final output.

MILESTONE ACTIVITYPLAN



SPRINT PLANING

Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story Points	Priority
Sprint – 1	Dataset Collection	USN – 1	Collect Dataset for building model	9	High
Sprint – 1	Image Preprocessing	USN – 2	Perform pre-processing techniques on thedataset	8	Medium
Sprint – 2	Model Building	USN – 3	Import the required libraries, add the necessary layersand compile the model	10	High
Sprint – 2		USN – 4	Training the image classification modelusing CNN	7	Medium
Sprint – 3	Training and Testing the Model	USN – 5	Training the model and testing the model's performance	9	High
Sprint – 4	Application Developme nt	USN – 6	Converting the input gesture image into English Alphabets	8	Medium

6.2 Sprint Delivery Schedule

Sprint	Total StoryPoints	Duration	Sprint StartDate	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint – 1	17	6 Days	24 October, 2022	29 October, 2022	17	29 Oct 2022
Sprint – 2	17	6 Days	31 October, 2022	05 November, 2022	17	05 Nov 2022
Sprint – 3	9	6 Days	07 November, 2022	12 November, 2022	9	12 Nov 2022
Sprint – 4	5	6 Days	14 November, 2022	19 November, 2022	8	19 Nov 2022

Velocity

Average Velocity= Velocity

Sprint Duration

- Average Velocity → AV
- Velocity → Points per sprint
- Sprint Duration → Number of days per sprint

1. Sprint – 1: AV = 17÷6 = 2.83
2. Sprint – 2: AV = 17÷6 = 2.83 '
3. Sprint – 3: AV = 9÷6 = 1.5
4. Sprint – 4: AV = 5÷6 = 0.83

6.3 Report From Jira

BURNDOWN CHART

CHAPTER 7

CODING AND EXECUTION

7.1 Feature 1

The proposed system consists of two features front end and backend. The frontend is designed using HTML and CSS. The first feature is a webpage whenever a user wants to translate the sign language to English, they can go to the webpage it has start button. On pressing the start button, it will turn on the camera for live translation. Once the camera is turned on, we can start translating.

Coding:

```
<!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>REAL TIME COMM</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="static/Navbar-Centered-Brand.css">
</head>

<body style="background: #6a6767;">
   <nav class="navbar navbar-light navbar-expand-md py-3" style="background: #FC3D3D;">
      <div class="container">
         <div></div><a class="navbar-brand d-flex align-items-left" href="#"><span
               class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex justify-content-left align-items-left me-
2 bs-icon"><i
                  class="fas fa-flask"></i></span><h4 style="color: #030000; font-style: oblique; text-align:
left;"><strong> Real-Time Communication
               System Powered By AI&amp;nbsp;For Specially Abled</strong></h4></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: 800px;height: 600px;margin: 10px;min-height: 480px;min-width: 640px;border-
radius: 50px;border: 10px groove #045816 ;">
            <img src="{{ url_for('video') }}" 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:
20px;"><button
            class="btn btn-info" type="button" data-bs-target="#modal-1" data-bs-toggle="modal"
            style=" background: #FC3D3D";>Quick Reference
            -<strong> ASL Alphabets</strong></button></div>
   </section>
<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='ASL_Alphabet.jpg' height=100% width="450px"></div>
            <div class="modal-footer"><button class="btn btn-secondary" type="button"
                  data-bs-dismiss="modal">Close</button></div>
         </div>
      </div>
   </div>
</section>
   <section>
      <div class="container">
         <div class="accordion text-white" role="tablist" id="accordion-1">
            <div class="accordion-item" style="font-style: oblique; 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="font-style:inherit; background: #FC3D3D;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">In our society, we have people with disabilities. The technology is
developing day by day but no significant developments are undertaken for the betterment of these people.
Communications between deaf-mute and a normal person has always been a challenging task. It is very
difficult for mute people to convey their message to normal people. Since normal people are not trained on
hand sign language. In emergency times conveying their message is very difficult. The human hand has
remained a popular choice to convey information in situations where other forms like speech cannot be used.
Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper
conversation between a normal person and an impaired person in any language..</p>
                  </div>

               </div>
            <div class="accordion-item" style="font-style: oblique; 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="font-style: oblique; background: #FC3D3D;color: rgb(231,241,255);">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 From Dr.MCET COLLEGE OF ENGINEERING AND
TECHNOLOGY<br><br>TEAM ID-- <strong>PNT2022TMID08592</strong><br><br>1.
<strong>PRAVEEN KUMAR S</strong> <br>2.
                     <strong>SAKTHI KAVIN K</strong> <br>3. <strong>LOGESH KUMAR
S</strong><br>4. <strong>SHIVAPRIYAN K</strong>
                  </p>
               </div>
            </div>
         </div>
      </div>
   </section>

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>

</html>
```

7.2 Feature 2

The second feature of the proposed system is backend. The backend is designed using python with the packages of python like flask, tensorflow, opencv-python, keras, numpy, pandas, virtualenv, pillow and Machine learning technology and trained with datasets. Once the camera is turned on the system detects and identify the sign language and translate it to English by matching the live action with the trained dataset.

Coding:

```
from flask import Flask, Response, render_template
from camera import Video
import cv2
from keras.models import load_model
from keras.preprocessing import image
import numpy as np
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import math

app = Flask(__name__)
camera = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1)
classifier = Classifier('E:/ibm project/prj10/keras_model.h5', 'E:/ibm project/prj10/labels.txt')
offset = 20
imgSize = 300

#folder = "Data/C"
counter = 0
```

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


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


def gen():
    while True:
        success, img = camera.read()
        imgOutput = img.copy()
        hands, img = detector.findHands(img)
        if hands:
            hand = hands[0]
            x, y, w, h = hand['bbox']

            imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
            imgCrop = img[y - offset:y + h + offset, x - offset:x + w + offset]

            imgCropShape = imgCrop.shape

            aspectRatio = h / w

            if aspectRatio > 1:
                k = imgSize / h
                wCal = math.ceil(k * w)
                imgResize = cv2.resize(imgCrop, (wCal, imgSize))
                imgResizeShape = imgResize.shape
                wGap = math.ceil((imgSize - wCal) / 2)
                imgWhite[:, wGap:wCal + wGap] = imgResize
                prediction, index = classifier.getPrediction(imgWhite, draw=False)
                print(prediction, index)

            else:
                k = imgSize / w
                hCal = math.ceil(k * h)
                imgResize = cv2.resize(imgCrop, (imgSize, hCal))
                imgResizeShape = imgResize.shape
                hGap = math.ceil((imgSize - hCal) / 2)
                imgWhite[hGap:hCal + hGap, :] = imgResize
                prediction, index = classifier.getPrediction(imgWhite, draw=False)

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

            # cv2.imshow("ImageCrop", imgCrop)
```

```
            # cv2.imshow("ImageWhite", imgWhite)

        #  cv2.imshow("Image", imgOutput)
          k = cv2.waitKey(1)
          if k == ord('q'):
              break

    camera.release()
    cv2.destroyAllWindows()

    yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + imgOutput + b'\r\n')



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


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

CHAPTER 8

TESTING

```
        # Importing Libarries

        from tensorflow.keras.models import load_model

        from tensorflow.keras.preprocessing import image

        import numpy as np

        import cv2

        # loading model

        model = load_model('E:\ibm

        project\Real time communication

        system\Application Building\Build a

        Flask Application\realtime.h5')

         from skimage.transform import resize
```

```
def detect(frame):
    img = resize(frame, (64, 64, 3))
    img = np.expand_dims(img, axis = 0)
    if np.max(img) > 1:
        img = img/255.0
    prediction = model.predict(img)
    print(prediction)
    return prediction
frame = cv2.imread(r"E:\ibm project\Real time communication
system\Dataset\test_set\A\16.png")
data = detect(frame)
index = ['A','B','C','D','E','F','G','H','I']
index[np.argmax(data)]
# Importing Libraries
import cv2
```

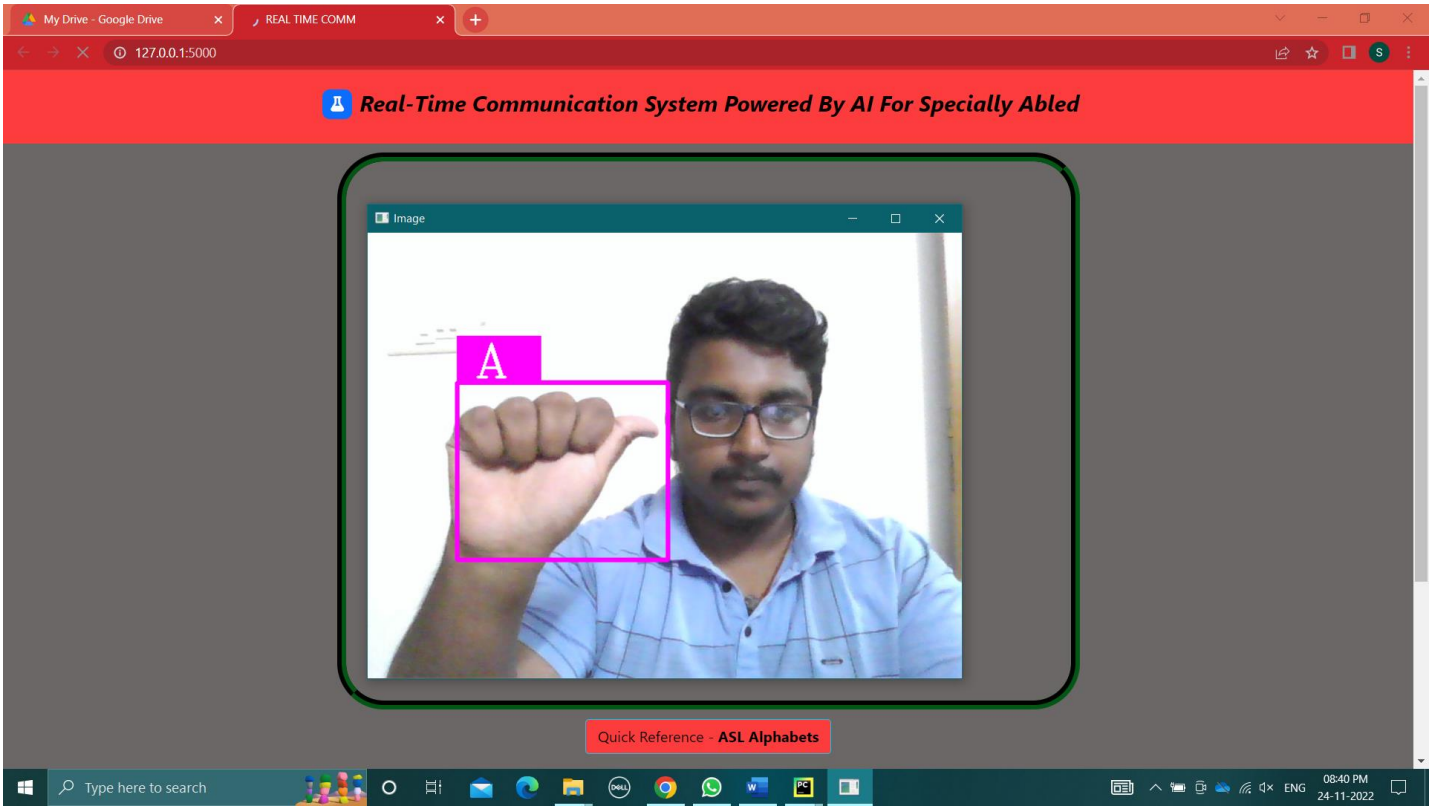
```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
# Loading Model
model = load_model("E:\ibm
project\Real time communication
system\Application Building\Build
a Flask Application\realtime.h5")
video = cv2.VideoCapture(0)
index = ['A','B','C','D','E','F','G','H','I']
while True:
    success, frame = video.read()
    cv2.imwrite('frame.jpg', frame)
    img = image.load_img('frame.jpg', target_size = (64, 64))
    x = image.img_to_array(img)
    x = cv2.cvtColor(x, cv2.COLOR_BGR2HSV)
    a = x.array_to_img(x)
    cv2.imshow("")
    x = np.expand_dims(x, axis = 0)
    pred = np.argmax(model.predict(x), axis = 1)
    y = pred[0]
    copy = frame.copy()
    cv2.rectangle(copy, (320, 100), (620, 400), (255, 0, 0), 5)
    cv2.putText(frame, "The Predicted Alphabet : " + str(index[y]), (100, 100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)
    cv2.imshow('frame', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

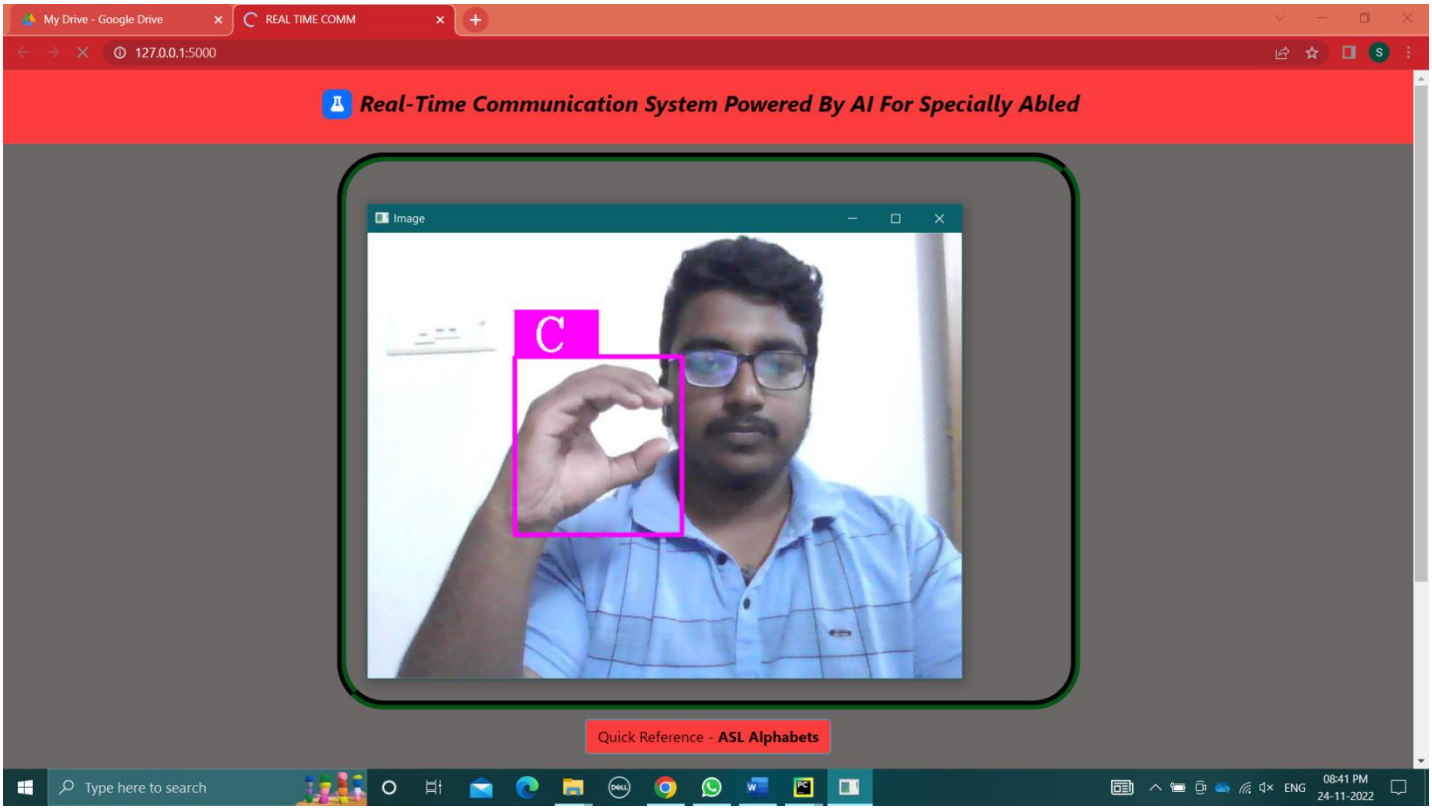
```
video.release()
cv2.destroyAllWindows()
```

CHAPTER 9

RESULT

9.1 Performance Metrics




My Drive - Google Drive
REAL TIME COMM
127.0.0.1:5000
Real-Time Communication System Powered By AI For Specially Abled
Image
C
Quick Reference - ASL Alphabets
Type here to search
ENG
08:41 PM
24-11-2022

CHAPTER 10

ADVANTAGE AND DISADVANTAGE

ADVANTAGE:

- Communication is the key in this society people with disability tends suffer but the proposed system provides a solution to them.
- Makes the translation of sign language to English easy.
- It can identify and translate the live and moving images.
- The proposed system ensures the easy translation of sign language to English.
- Even the people with lack of sign language can use the proposed system easily.
- This does not require high-end device to use it.
- Can be used on almost all operating systems and browses.
- Does not require prior programming knowledge t use the system
- The proposed system is user friendly.
- Makes the life of the person with disability easy.

DISADVANTAGE:

- The proposed system is not a two-way translation system.
- There is chance for wrong translation.
- Since it is a webpage-based system, it does require internet connectivity which can be inconvenient at times.
- It would have been convenient if it is application based.

CHAPTER 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 recognizes 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.

CHAPTER 12

FUTURE SCOPE

In the future to take the project to the next level two way communication system such as sign language to english and english to sign language is beign under the planning phase.The application version of the web page for both ios and android is also in planning process for the future development.Research to improve the accuracy of the system is under progress.

CHAPTER 13

APPENDIX

SOURCE CODE:

HTML:

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

<head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
   <title>REAL TIME COMM</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="static/Navbar-Centered-Brand.css">
</head>

<body style="background: #6a6767;">
   <nav class="navbar navbar-light navbar-expand-md py-3" style="background: #FC3D3D;">
      <div class="container">
         <div></div><a class="navbar-brand d-flex align-items-left" href="#"><span
               class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex justify-content-left align-items-left me-
2 bs-icon"><i
                  class="fas fa-flask"></i></span><h4 style="color: #030000; font-style: oblique; text-align:
left;"><strong> Real-Time Communication
               System Powered By AI&amp;nbsp;For Specially Abled</strong></h4></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: 800px;height: 600px;margin: 10px;min-height: 480px;min-width: 640px;border-
radius: 50px;border: 10px groove #045816 ;">
            <img src="{{ url_for('video') }}" 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:
20px;"><button
            class="btn btn-info" type="button" data-bs-target="#modal-1" data-bs-toggle="modal"
            style=" background: #FC3D3D";>Quick Reference
            -<strong> ASL Alphabets</strong></button></div>
   </section>
<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='ASL_Alphabet.jpg' height=100% width="450px"></div>
          <div class="modal-footer"><button class="btn btn-secondary" type="button"
              data-bs-dismiss="modal">Close</button></div>
        </div>
      </div>
    </div>
</section>
    <section>
      <div class="container">
        <div class="accordion text-white" role="tablist" id="accordion-1">
          <div class="accordion-item" style="font-style: oblique; 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="font-style:inherit; background: #FC3D3D;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">In our society, we have people with disabilities. The technology is
developing day by day but no significant developments are undertaken for the betterment of these people.
Communications between deaf-mute and a normal person has always been a challenging task. It is very
difficult for mute people to convey their message to normal people. Since normal people are not trained on
hand sign language. In emergency times conveying their message is very difficult. The human hand has
remained a popular choice to convey information in situations where other forms like speech cannot be used.
Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper
conversation between a normal person and an impaired person in any language..</p>
              </div>

            </div>
          <div class="accordion-item" style="font-style: oblique; 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="font-style: oblique; background: #FC3D3D;color: rgb(231,241,255);">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 From Dr.MCET COLLEGE OF ENGINEERING AND
TECHNOLOGY<br><br>TEAM ID-- <strong>PNT2022TMID08592</strong><br><br>1.
<strong>PRAVEEN KUMAR S</strong> <br>2.
                  <strong>SAKTHI KAVIN K</strong> <br>3. <strong>LOGESH KUMAR
S</strong><br>4. <strong>SHIVAPRIYAN K</strong>
                </p>
```

```
                </div>
            </div>
        </div>
    </div>
</div>
</section>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>

</html>
```

PYTHON:

```
from flask import Flask, Response, render_template
from camera import Video
import cv2
from keras.models import load_model
from keras.preprocessing import image
import numpy as np
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import math

app = Flask(__name__)
camera = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1)
classifier = Classifier('E:/ibm project/prj10/keras_model.h5', 'E:/ibm project/prj10/labels.txt')
offset = 20
imgSize = 300

#folder = "Data/C"
counter = 0

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


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


def gen():
    while True:
        success, img = camera.read()
        imgOutput = img.copy()
        hands, img = detector.findHands(img)
        if hands:
            hand = hands[0]
            x, y, w, h = hand['bbox']
```

```
        imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
        imgCrop = img[y - offset:y + h + offset, x - offset:x + w + offset]

        imgCropShape = imgCrop.shape

        aspectRatio = h / w

        if aspectRatio > 1:
            k = imgSize / h
            wCal = math.ceil(k * w)
            imgResize = cv2.resize(imgCrop, (wCal, imgSize))
            imgResizeShape = imgResize.shape
            wGap = math.ceil((imgSize - wCal) / 2)
            imgWhite[:, wGap:wCal + wGap] = imgResize
            prediction, index = classifier.getPrediction(imgWhite, draw=False)
            print(prediction, index)

        else:
            k = imgSize / w
            hCal = math.ceil(k * h)
            imgResize = cv2.resize(imgCrop, (imgSize, hCal))
            imgResizeShape = imgResize.shape
            hGap = math.ceil((imgSize - hCal) / 2)
            imgWhite[hGap:hCal + hGap, :] = imgResize
            prediction, index = classifier.getPrediction(imgWhite, draw=False)

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

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

    #  cv2.imshow("Image", imgOutput)
      k = cv2.waitKey(1)
      if k == ord('q'):
          break

    camera.release()
    cv2.destroyAllWindows()

    yield (b'--frame\r\n'
           b'Content-Type: image/jpeg\r\n\r\n' + imgOutput + b'\r\n')



@app.route('/video')
def video():
```

```
    # video = Video()
    return Response(gen(), mimetype='multipart/x-mixed-replace; boundary=frame')


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

TRAINNING CODE:

```
    # Importing Libraries

    from tensorflow.keras.preprocessing.image import ImageDataGenerator

    # Image Augmentation

    train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range =
    0.2, horizontal_flip = True)

    test_datagen = ImageDataGenerator(rescale = 1./255)

    # Loading train and test set

    X_train = train_datagen.flow_from_directory(r"E:\ibm project\Real time communication
    system\Dataset\training_set", target_size = (64, 64), batch_size = 32, class_mode
    = 'categorical')

    X_test = test_datagen.flow_from_directory(r"E:\ibm project\Real time communication
    system\Dataset\training_set", target_size = (64, 64), batch_size = 32, class_mode
    = 'categorical')

    # checking indices

    X_train.class_indices

    # Importing Libraries

    from tensorflow.keras.models import Sequential

    from tensorflow.keras.layers import Dense
```

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

# Initializing the Model

model = Sequential()

# Adding Convolution Layer

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

# Adding Pooling Layer

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

# Adding Flatten Layer

model.add(Flatten())

# Adding Hidden Layer

model.add(Dense(units = 512, kernel_initializer = 'random_uniform', activation = 'relu'))

# Adding Output Layer

model.add(Dense(units = 9, kernel_initializer = 'random_uniform', activation =
'softmax'))

# Compile the model

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

# Fiiting the model

model.fit_generator(X_train, steps_per_epoch = 24, epochs = 10, validation_data =
X_test, validation_steps = 40)

# Saving the model

model.save('aslpng1.h5')
```

TESTING CODE:

```
# Importing Libarries

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image
```

```
import numpy as np
import cv2
# loading model
model = load_model('realtime.h5')
from skimage.transform import resize
def detect(frame):
    img = resize(frame, (64, 64, 3))
    img = np.expand_dims(img, axis = 0)
    if np.max(img) > 1:
        img = img/255.0
    prediction = model.predict(img)
    print(prediction)
    return prediction
frame = cv2.imread(r"E:\ibm project\Real time communication
system\Dataset\test_set\A\16.png")
data = detect(frame)
index = ['A','B','C','D','E','F','G','H','I']
index[np.argmax(data)]
# Importing Libraries
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
# Loading Model
model = load_model("real time.h5")
```

```
video = cv2.VideoCapture(0)
index = ['A','B','C','D','E','F','G','H','I']
while True:
    success, frame = video.read()
    cv2.imwrite('frame.jpg', frame)
    img = image.load_img('frame.jpg', target_size = (64, 64))
    x = image.img_to_array(img)
    x = cv2.cvtColor(x, cv2.COLOR_BGR2HSV)
    a = x.array_to_img(x)
    cv2.imshow("")
    x = np.expand_dims(x, axis = 0)
    pred = np.argmax(model.predict(x), axis = 1)
    y = pred[0]
    copy = frame.copy()
    cv2.rectangle(copy, (320, 100), (620, 400), (255, 0, 0), 5)
    cv2.putText(frame, "The Predicted Alphabet : " + str(index[y]), (100, 100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)
    cv2.imshow('frame', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

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

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-2066-1658425644

DEMO LINK:

https://drive.google.com/file/d/1yj986G_ZND1POuynnNlDB6jH9f0COU_y/view?usp=share_link