Real-Time Communication System Powered by AI for Specially Abled

SUBMITTED BY

JEEVITHA PALANISAMY (921019106013)

SHUBAKEERTHI MALAICHAMY (921019106035)

MUTHULAKSHMI KUMARESAN (921019106023)

SATHYA MUTHUPANDI (921019106032)

GUIDED BY

Dr.R.ATHILINGAM

For

Professional Readiness for Innovation, Employability and Entrepreneurship

(HX8001)

Department of Electronics and Communication Engineering

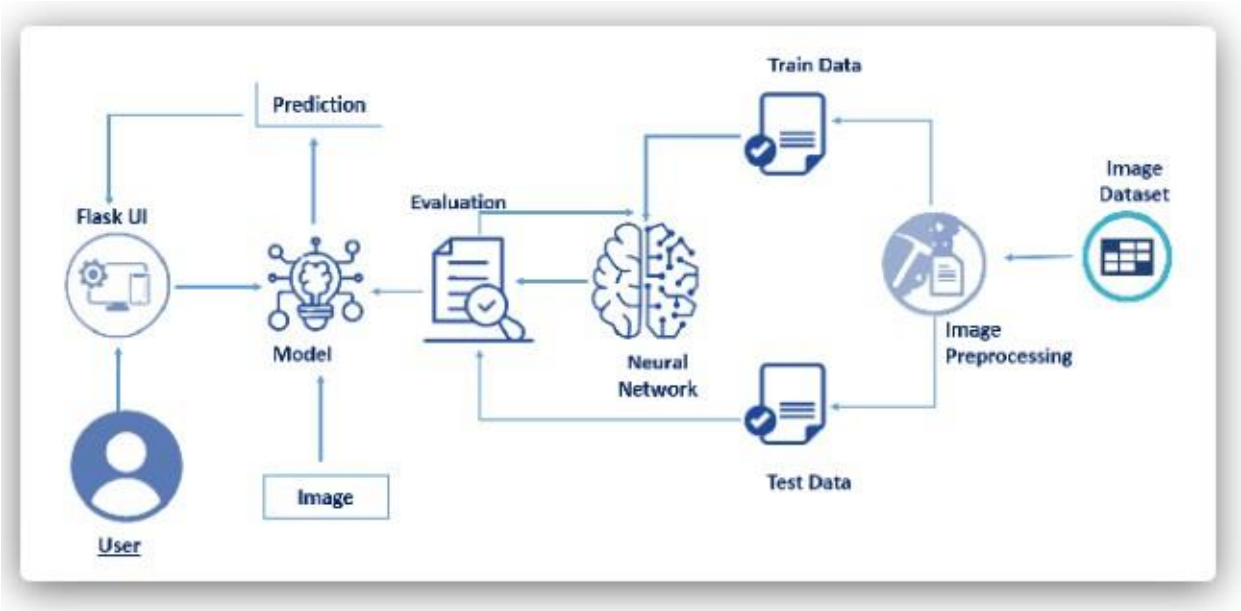
Nadar Saraswathi College of Engineering and Technology

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. 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.

Technical Architecture:



1.2 Purpose

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.

2. LITERATURE SURVEY

Survey on Real-Time Communication System Powered By AI for Specially Abled

TITLE	AUTHOR	YEAR	THEME	IDEA
full duplex intelligent communication system for deaf and dumb people	Surbhi Rathi Ujwalla Gawande	2017	intelligent communication system for deaf and dumb people	make a dual way **communication** system between **deaf dumb** and normal people my full duplex system first the recognized gesture has converted into text message and voice format, so that normal person can understand it. Second, voice has converted into text message and its corresponding gesture, so that physically impaired humans can understand it.
Multimodal Interface for Deaf and Dumb Communication	Wazalwar, Sampada S. and Shrawankar, Urmila	2019	Creating a multimodal interface between deaf, dump and normal people	Deaf & Dumb can use multimodal interfaces for communication such as image, video, Leap reading, air writing, finger spelling & body gestures etc. Everyone is unable to learn sign language efficiently thus multimodal interface can serve the purpose of communication.Ξ
Image-based Bengali Sign Recognition for Deaf and Dumb	Rafi, Abdul Muntakim and Nawal, Nowshin and Bayev, Nur Sultan Nazar	2019	relies on the images of bare hands, which allows the users to interact with the system in a natural way.	They have collected in total 12581 different hand signs for the 38 BdSL alphabets in collaboration with the National Federation of the Deaf. They proposed a VGG19 based convolutional neural network for the recognition of 38 classes

				and achieve an overall test accuracy of 89.6%.
Sign Language Recognition for Deaf and Dumb People Using Android Environment	A.Gayathiri A.Sasi kumar	2017	Android application development powered by VRS that translates spoken and written words into sign language	Speech-to-sign technology and VRS enables audible language translation on smart phones with signing and application has characters feature in mobile without dialing number uses a technology that translates spoken and written words into sign language with video.
Assistive Sign Language Converter for Deaf and Dumb	Boppana, Lakshmi and Ahamed, Rasheed and Rane, Harshali and Kodali, Ravi Kishore	2019	This device allows the person to communicate with sign hand postures in order to recognize different gestures based signs.	The controller of this assistive device is developed for processing the images of gestures by employing various image processing techniques and deep learning models to recognize the sign. This sign is converted into speech in real-time using text-to-speech module.

Survey on Real-Time Communication System Powered By AI for Specially Abled

Hand Gesture Recognition for Deaf and Dumb Using CNN Technique	Vanaja, S. and Preetha, R. and Sudha, S.	2021	A Hand gesture recognition system to aid deaf and mute is developed using convolutional neural networks to identify the static signs of ISL (Indian Sign Language)	A total of 4 layers and 16 filters were used in Convolution Neutral Network (CNN) Architecture based on Deep learning technique. Adam optimizer has been used as the optimizer to tweak the weighs of the model is useful for reducing the loss and improving the accuracy. Model is trained in total of 15 epochs. The optimizer used to train and validate process is Stochastic Gradient Descent (SGD). The proposed model gives the maximum possible training accuracy of about 99.76%.
Hand Gesture Recognition and Voice Conversion for Deaf and Dumb	Rupesh Prajapati1, Vedant Pandey2, Nupur Jamindar3, Neeraj Yadav	2018	Conversion of recognized image which was taken from a normal person who tries to interact with deaf people .convert this information into voice or text for deaf and dump	provides a powerful tool for data analysis and pattern recognition which is often used in signal and image processing as a technique for data compression, data dimension reduction or their decorrelation as well.

Survey on Real-Time Communication System Powered By AI for Specially Abled

			people	
Real time sign language interpreter	Nath, Geethu G and Arun, C S	2017	applications including gesture-controlled activities like human computer interaction, gesture-controlled home appliances and other electronic devices and many applications that uses gesture as the trigger input.	system for sign language recognition for deaf and dumb people is implemented in ARM CORTEX A8 processor board using convex hull algorithm and template matching algorithm. Image is obtained using webcam. This hand sign image is converted to text so as to develop a communication between normal and deaf and dumb people. Open CV is the software tool that provides the support with image processing techniques. The system converts sign language to text for deaf and dumb people to communicate with normal people
American Sign Language Recognition System: An Optimal Approach	Shivashankara S Srinath S	2018	ASL provides a set of 26 gesture signs named as an American Manual Alphabet that can be cast-off to spell out many of the	Pre-processing operations of the signed input gesture are done in the first phase. In the next phase, the various region properties of pre-processed gesture image is computed. In the final phase, based on the properties calculated of earlier phase, the

Survey on Real-Time Communication System Powered By AI for Specially Abled

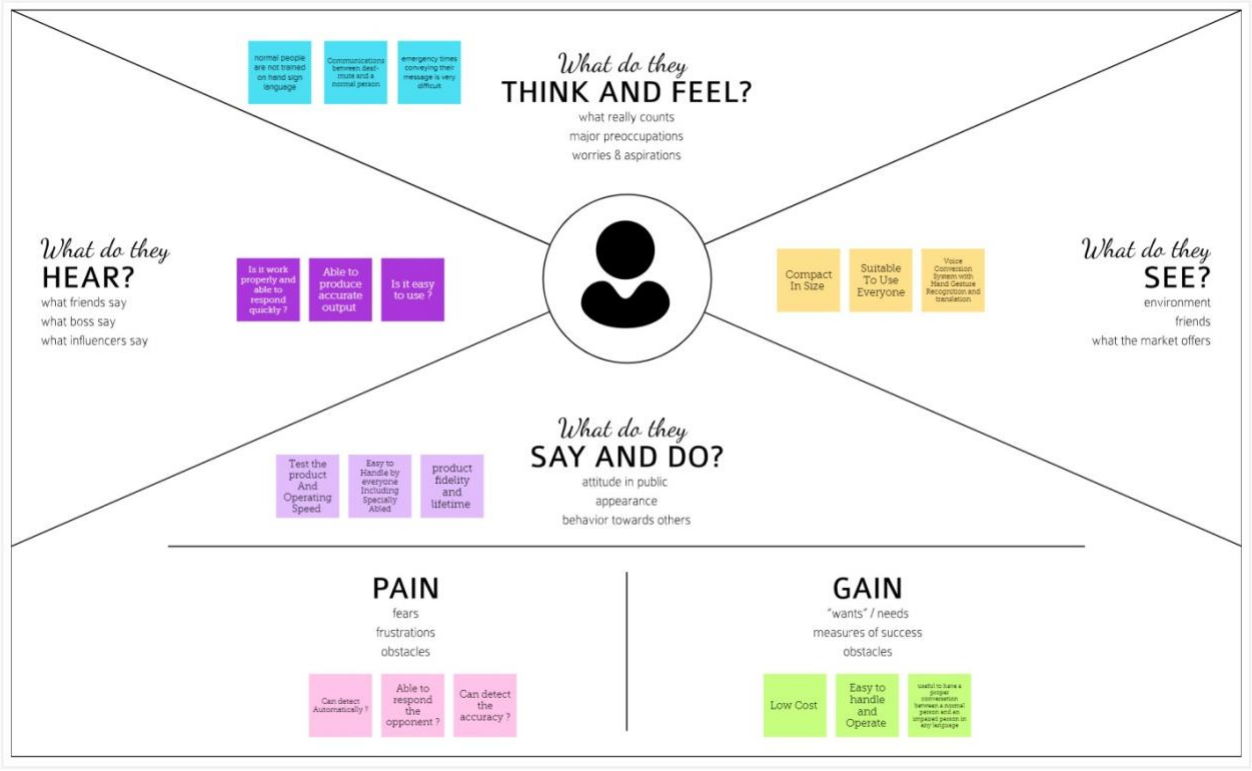
			English words available. The 19 various hand shapes of ASL are cast-off to make 26 American Manual Alphabets. An identical hand shape with diverse orientations is used for 'K' and 'P' letters signs. In ASL, also offers a set of 10 numeric gestures to sign the numbers '0' to '9'. ASL doesn't comprises built-in ASL equivalents signs for accurate nouns and technical terms.	transliteration of signed gesture into text has been carried out The sign language (SL) is made by specifications of hand and facial idioms to express their views and thoughts of speech and hearing disabled persons with the normal (speech and hearing) people. Most of the normal persons may not clearly understand the sign language. Therefore, there is a massive communication gap between the deaf communities with the general public

Reference link: Project Design & Planning/Ideation Phase/Literature survey.pdf

https://github.com/IBM-EPBL/IBM-Project-39684-1660488681/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/IDEATION%20PHASE/LITERATURE%20%20%20%20SURVEY.docx

IDEATION & PROPOSED SOLUTION

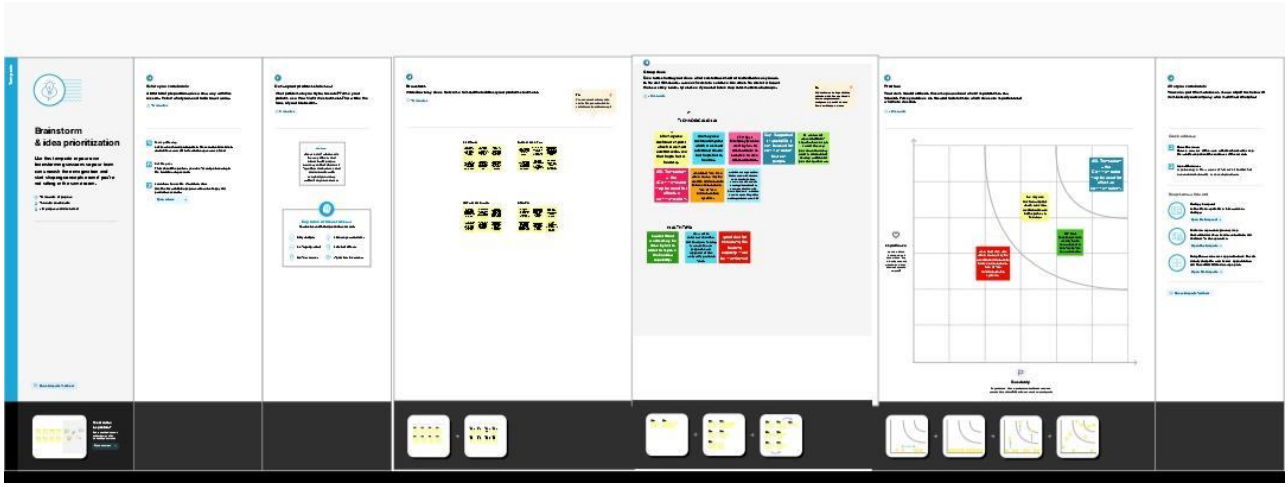
2.4 Empathy Map Canvas



Reference link: Project Design & Planning/Ideation Phase/Empathy Map.pdf

https://github.com/IBM-EPBL/IBM-Project-39684-1660488681/blob/main/empathy%20map.png

2.5 Ideation & Brainstorming



Reference link: Project Design & Planning/Ideation Phase/Brainstorm & idea prioritization.pdf

https://github.com/IBM-EPBL/IBM-Project-39684-1660488681/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/IDEATION%20PHASE/BRAINSTROM%20%20AND%20IDEA%20PRIORIZATION.pdf

2.6 Proposed Solution

Project Design Phase-I
Proposed Solution Template

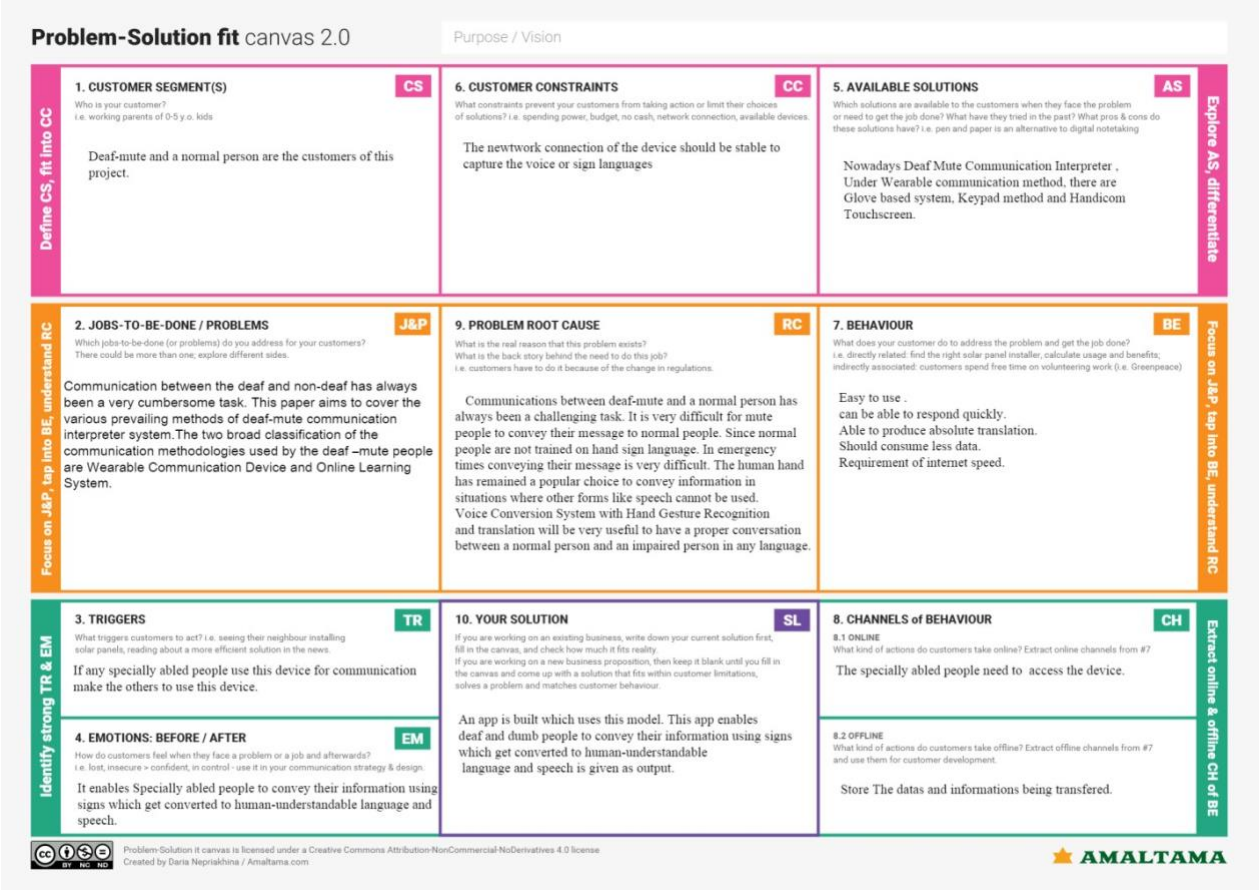
Date	19 September 2022
Team ID	PNT2022TMID48841
Project Name	Project - Real-Time Communication System Powered By AI For Specially Abled
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Communications between deaf-mute and a normal person.
2.	Idea / Solution description	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.
3.	Novelty / Uniqueness	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.
4.	Social Impact / Customer Satisfaction	An App is used that enables deaf-mute people to convey their information using signs which get converted to human Understandable language and speech.
5.	Business Model (Revenue Model)	An app is being built which uses this model.
6.	Scalability of the Solution	Can use both normal and deaf-mute people. Easy to handle. Produces rapid translations. Deliver the accurate content.

Reference link: https://github.com/IBM-EPBL/IBM-Project-39684-1660488681/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/PHASE%201/PROPOSED%20SOLUTIONS.pdf

2.7 Problem Solution fit



Reference link: https://github.com/IBM-EPBL/IBM-Project-39684-1660488681/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/PHASE%201/PROBLEM%20SOLUTION%20FIT.pdf

3. REQUIREMENT ANALYSIS

3.1 Functional requirement

3.2 Non-Functional requirements

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	15 October 2022
Team ID	PNT2022TMID48853
Project Name	Project - Real-Time Communication System Powered By AI For Specially Abled
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Collect data set	Open the web page in google chrome and collect the image.
FR-2	Image uploading	Uploading the data in web page.
FR-3	Access the data	Access the trained data in the code.
FR-4	Using webcam/ camera/ voice receiver	Collect the input.
FR-5	Display	Produce converting output.

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Easy to Handle.
NFR-2	**Security**	Produces output when a voice or sign language is given as an input.
NFR-3	**Reliability**	Able to identify the speech and sign input and produces an output.
NFR-4	**Performance**	Rapid response while converting. Produce accurate output.
NFR-5	**Availability**	Nowadays Deaf Mute Communication Interpreter, Under Wearable communication method, there are Glove based system, Keypad method and Handicom Touchscreen.
NFR-6	**Scalability**	Easy to use. Can be able to respond quickly. Able to produce absolute translation. Should consume less data. Requirement of internet speed.

Reference Link: https://github.com/IBM-EPBL/IBM-Project-39684-1660488681/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/PHASE%202/SOLUTION%20REQIREMENTS%20JEEVITHA.pdf

PROJECT DESIGN

3.3 Data Flow Diagrams

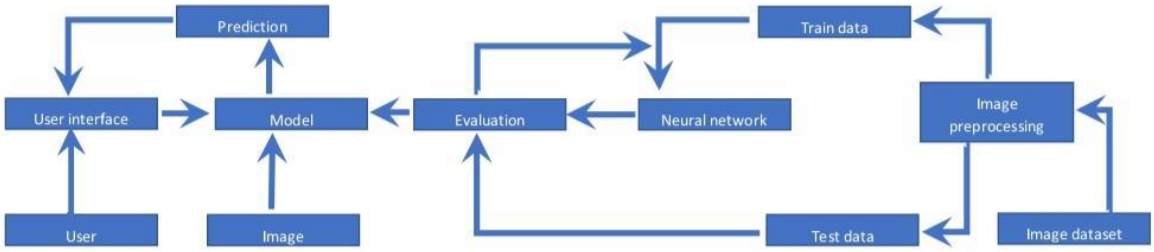
3.4 Solution & Technical Architecture

3.5 User Stories

Project Design Phase-II
Data Flow Diagram & User Stories

Date	15 October 2022
Team ID	PNT2022TMID48853
Project Name	Project - Real-Time Communication System Powered By AI For Specially Abled
Maximum Marks	4 Marks

Data Flow Diagrams:



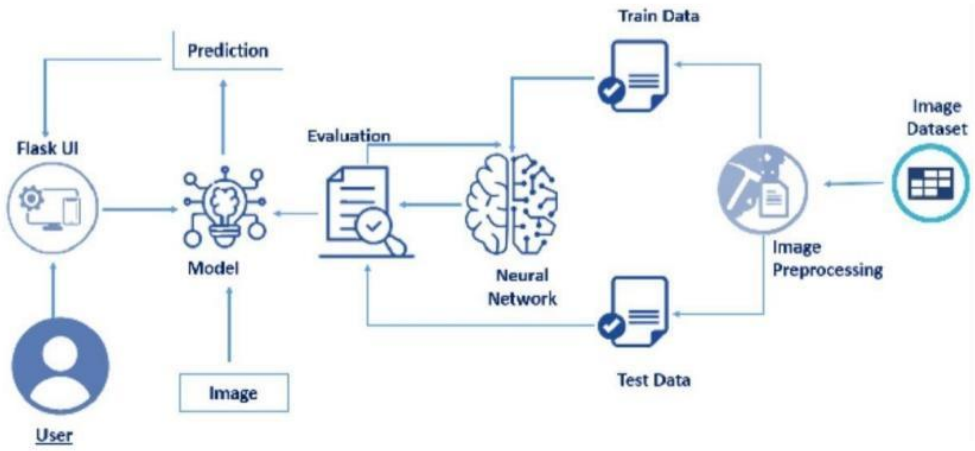
User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Open application	USN-1	Open the application with a help of web page or mobile application.	Can open in app/chrome	High	Sprint-1
	Home page	USN-2	Link directed into home page.	-	High	Sprint-1
	Introduction page	USN-3	Click on the demo/ introduction.	Introduction page will open. Follow the instructions given.	Medium	Sprint-2
	Launch application	USN-4	Click launch to move the next page.	Launch the application, it will redirected to the next page.	Medium	Sprint-1
	Selecting the conversion	USN-5	User need to select the conversion	User should select the conversion from text to sign or sign to text.	High	Sprint-1
	Output / conversion	USN - 6	Output on regional language	The gesture or text will display		
Customer (Web user)	same for both users					
Customer Care Executive	Same for both normal and specially abled people					
Administrator	same for all the users					

Project Deslgn Phase-II
Technology Stack (Architecture & Stack)

Technical Architecture:

Trades € s



Referenc Link: https://github.com/IBM-EPBL/IBM-Project-39684-1660488681/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/PROJECT%20PLANNING%20PHASE/MILESTONE%20AND%20TASKS.pdf

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI	HTML, CSS, JavaScript.
2.	Data Set	Collect the data set consist of hand sign gesture.	From online
3.	Application Logic-1	Import all the library files required for data preprocessing.	Python
4.	Application Logic-2	Build the CNN model.	Python
5.	Application Logic-3	Login into Jupyter notebook.	Online or application download
6.	Data storage	Load / store the dataset and code.	System storage.
7.	Cloud Database	Database Service on Cloud	IBM Cloud
8.	Infrastructure (Server / Cloud)	Train the dataset and model using IBM cloud	IBM Cloud
9.	Machine Learning Model	Used to analyze visual images, image processing, video capture and analysis including features like face detection and hand sign detection.	CNN, Anaconda

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Application development, data pre-processing.	Pycharm , anaconda navigator
2.	Security Implementations	Produces an translation output when an speech or sign language is given as an input.	Anaconda
3.	Scalable Architecture	Easy to use. Can be able to respond quickly. Able to produce absolute translation. Should consume less data. Requirement of internet speed.	Anaconda
4.	Availability	Nowadays Deaf Mute Communication Interpreter, Under Wearable communication method, there are Glove based system, Keypad method and Handicom Touchscreen.	Artificial Intelligence
5.	Performance	Rapid conversion from sign language to text or text to sign language.	CNN Model

Reference Link: https://github.com/IBM-EPBL/IBM-Project-39684-1660488681/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/PHASE%202/TECHNOLOGY%20STACK%20%20JEEVITHA.pdf

4. PROJECT PLANNING & SCHEDULING

4.1 Sprint Planning & Estimation

4.2 Sprint Delivery Schedule

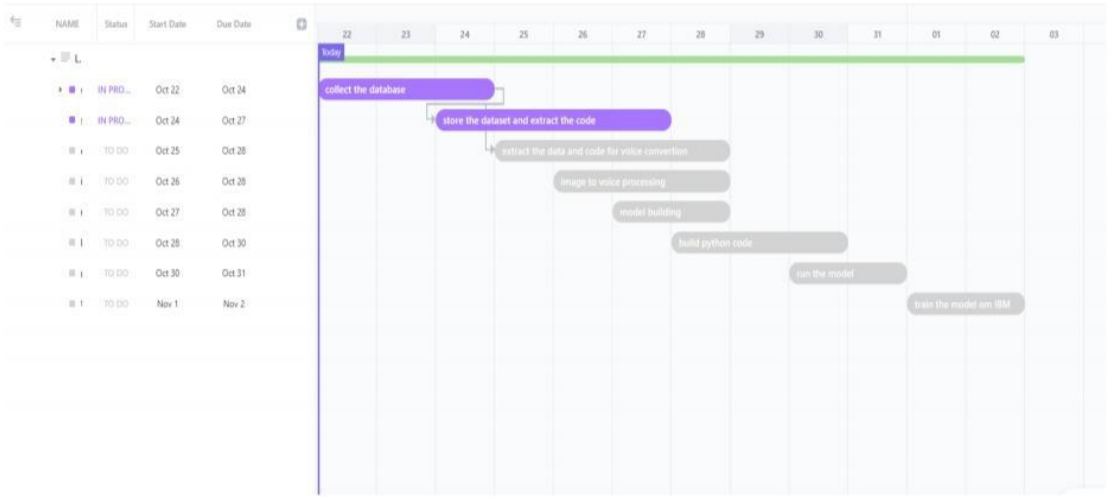
4.3 Reports from JIRA

PROJECT PLANNING PHASE

MILESTONE AND TASKS

DATE	22 October
TEAM ID	PNT2022TMID48853
PROJECT NAME	Real time communication Powered by AI for physically abled.
MAXIMUM MARKS	8 marks

Link for the below milestone: https://app.clickup.com/43289876/v/g/19938m-84



Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	22 October 2022
Team ID	PNT2022TMID48853
Project Name	Real time communication system Powered by AI for physically abled.
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint -1	Download the application	USN-1	As a user, download and open the application or website.	2	High	P.Jeevitha K.Muthu Lakshmi M. Shuba Keerthi M.sathya
Sprint -1		USN-2	The application will open, the camera is turned on for recording the sign languages	1	High	P.Jeevitha K.Muthu Lakshmi M. Shuba Keerthi M.sathya
Sprint -2		USN-3	The recorded language is taken as input	2	Low	P.Jeevitha K.Muthu Lakshmi M. Shuba Keerthi M.sathya
Sprint -1	Conversion process	USN-4	With the help of CNN in AI the sign language is converted into texts	2	Medium	P.Jeevitha K.Muthu Lakshmi M. Shuba Keerthi M.sathya
Sprint -1	Voice hearing	USN-5	The texts are then converted into audio for understanding	1	High	P.Jeevitha K.Muthu Lakshmi M. Shuba

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			of normal people.			Keerthi M.sathya

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

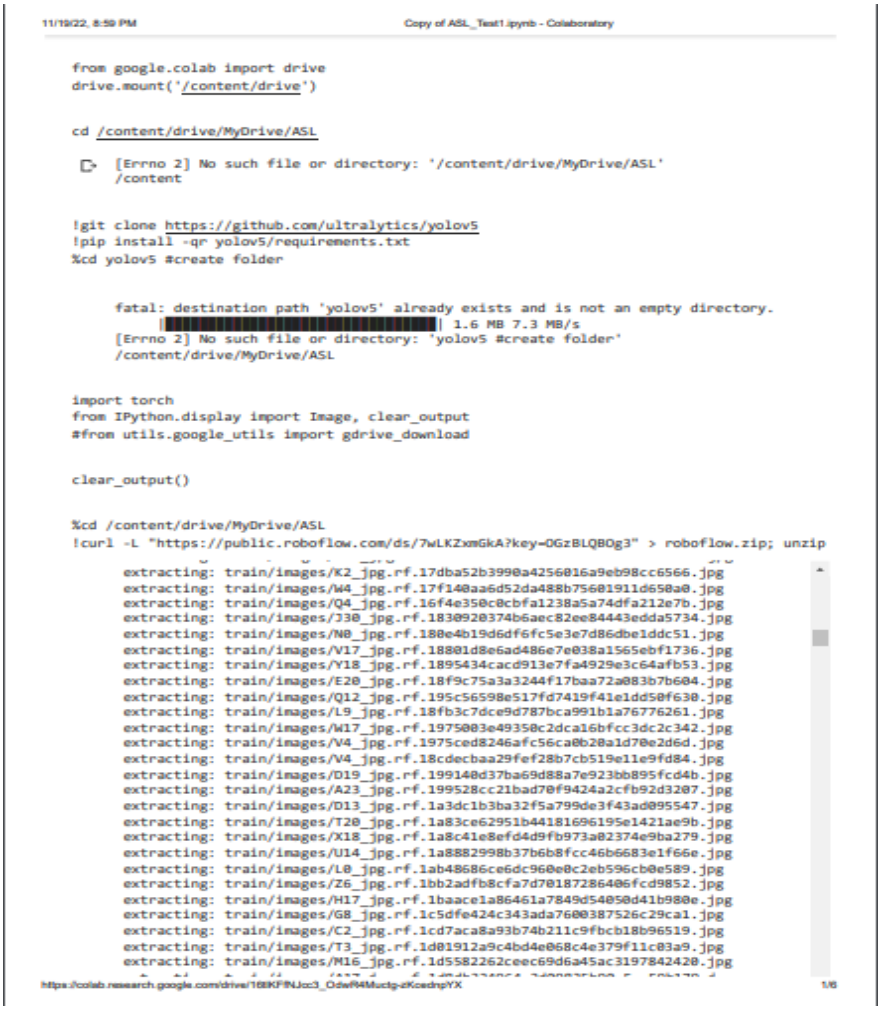
Reference Link: https://github.com/IBM-EPBL/IBM-Project-39684-1660488681/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/PROJECT%20PLANNING%20PHASE/SPRINT%20PLANNING%20AND%20SCHEDULES.pdf

5. CODING & SOLUTIONING (Explain the features added in the project along with code)

5.1 Feature 1

5.2 Feature 2

5.3 Database Schema (if Applicable)



11/19/22, 8:59 PM Copy of ASL_Test1.ipynb - Colaboratory

```
from google.colab import drive
drive.mount('/content/drive')

cd /content/drive/MyDrive/ASL

[Errno 2] No such file or directory: '/content/drive/MyDrive/ASL'
/content

!git clone https://github.com/ultralytics/yolov5
!pip install -qr yolov5/requirements.txt
%cd yolov5 #create folder

    fatal: destination path 'yolov5' already exists and is not an empty directory.
     |████████████████████████████████| 1.6 MB 7.3 MB/s
    [Errno 2] No such file or directory: 'yolov5 #create folder'
    /content/drive/MyDrive/ASL

import torch
from IPython.display import Image, clear_output
#from utils.google_utils import gdrive_download

clear_output()

%cd /content/drive/MyDrive/ASL
!curl -L "https://public.roboflow.com/ds/7wLKZxmGkA?key=OGzBLQBOg3" > roboflow.zip; unzip
      extracting: train/images/K2_jpg.rf.17dba52b3990a4256016a9eb98cc6566.jpg
      extracting: train/images/W4_jpg.rf.17f140aa6d52da488b75601911d650a0.jpg
      extracting: train/images/Q4_jpg.rf.16f4e350c0cbfa1238a5a74dfa212e7b.jpg
      extracting: train/images/J30_jpg.rf.1830920374b6aec82ee84443edda5734.jpg
      extracting: train/images/N0_jpg.rf.180e4b19d6df6fc5e3e7d86dbe1ddc51.jpg
      extracting: train/images/V17_jpg.rf.18801d8e6ad486e7e038a1565ebf1736.jpg
      extracting: train/images/Y18_jpg.rf.1895434cacd913e7fa4929e3c64afb53.jpg
      extracting: train/images/E20_jpg.rf.18f9c75a3a3244f17baa72a083b7b604.jpg
      extracting: train/images/Q12_jpg.rf.195c56598e517fd7419f41e1dd50f630.jpg
      extracting: train/images/L9_jpg.rf.18fb3c7dce9d787bca991b1a76776261.jpg
      extracting: train/images/W17_jpg.rf.1975003e49350c2dca16bfcc3dc2c342.jpg
      extracting: train/images/V4_jpg.rf.1975ced8246afc56ca0b20a1d70e2d6d.jpg
      extracting: train/images/V4_jpg.rf.18cdecbaa29fef28b7cb519e11e9fd84.jpg
      extracting: train/images/D19_jpg.rf.199140d37ba69d88a7e923bb895fcd4b.jpg
      extracting: train/images/A23_jpg.rf.199528cc21bad70f9424a2cfb92d3207.jpg
      extracting: train/images/D13_jpg.rf.1a3dc1b3ba32f5a799de3f43ad095547.jpg
      extracting: train/images/T20_jpg.rf.1a83ce62951b44181696195e1421ae9b.jpg
      extracting: train/images/X18_jpg.rf.1a8c41e8efd4d9fb973a82374e9ba279.jpg
      extracting: train/images/U14_jpg.rf.1a8882998b37b6b8fcc46b6683e1f66e.jpg
      extracting: train/images/L0_jpg.rf.1ab48686ce6dc960e0c2eb596cb0e589.jpg
      extracting: train/images/Z6_jpg.rf.1bb2adfb8cfa7d7018728640 6fcd9852.jpg
      extracting: train/images/H17_jpg.rf.1baace1a86461a7849d54050d41b980e.jpg
      extracting: train/images/G8_jpg.rf.1c5dfe424c343ada7600387526c29ca1.jpg
      extracting: train/images/C2_jpg.rf.1cd7aca8a93b74b211c9fbcb18b96519.jpg
      extracting: train/images/T3_jpg.rf.1d01912a9c4bd4e068c4e379f11c03a9.jpg
      extracting: train/images/M16_jpg.rf.1d5582262ceec69d6a45ac3197842420.jpg
```

https://colab.research.google.com/drive/16BKFINJcc3_OdwR4MuctgvKcednpYX 1/6

```
extracting: train/images/D3_jpg.rf.1d937c0e64141083b4b634f3b9a7eea9.jpg
```

```
            train/images/N10_jpg.rf.1e1d01dfd5fb83b817aa27f717ffef77.jpg
extracting: train/images/R4_jpg.rf.1e39e76f2b8933561c30caebc8b12fab.jpg
extracting: train/images/L15_jpg.rf.1e825d1bdfe88d7122179bc79e358eb1.jpg
extracting: train/images/M4_jpg.rf.1e94cac19b236e16f867f4292775a7a3.jpg
extracting: train/images/J35_jpg.rf.1ed39e38a0ee03a335348b9103c6121e.jpg
```

```
            train/images/C18_jpg.rf.1f4779c94ba51a17eb060430bfdefe84.jpg
extracting: train/images/D11_jpg.rf.1f53ba2b436994ad36b120c820921bdc.jpg
extracting: train/images/I20_jpg.rf.1f767dcc423ec7719b9e0e90f7ad3e42.jpg
extracting: train/images/A25_jpg.rf.1ee291da526c10ede77cf69d4cd2c231.jpg
extracting: train/images/M13_jpg.rf.1fa25f00496dd516c39840a417137f60.jpg
```

```
extracting: train/images/F22_jpg.rf.203feff812b6af7fb2f3c76ec919416d.jpg
extracting: train/images/C20_jpg.rf.205170f82acd823e1977ac5fa1102737.jpg
extracting: train/images/T22_jpg.rf.205c80aa55302bd71fe1c0ac89833f07.jpg
```

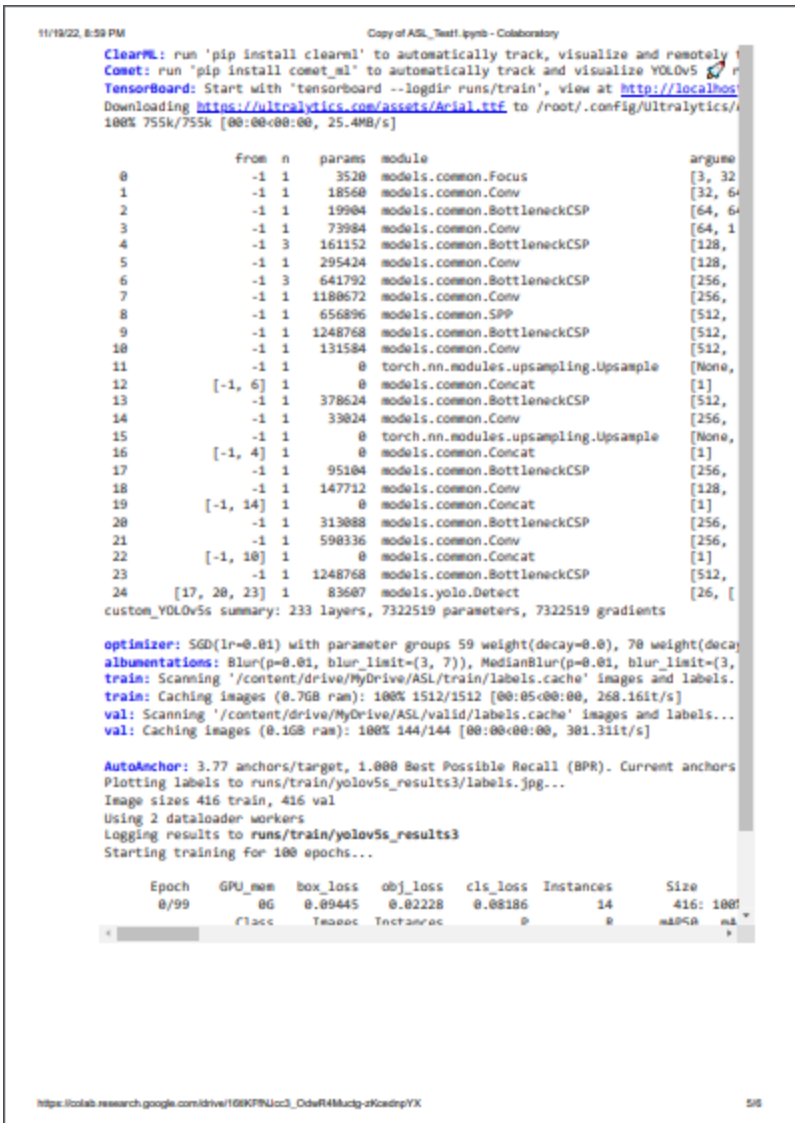
```python
with open("data.yaml", 'r') as stream:
    num_classes = str(yaml.safe_load(stream)['nc'])
```

dspc%_æłcMko: 0.fl W mool dmmth æAt*#Ls
æBdc%_æłc#elo: 0.SE door c ænolnlClpAs

```
    - [30,61, 62,45, 59,119]  # P4/16
    - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [64, 6, 2, 2]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 6, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 3, C3, [1024]],
   [-1, 1, SPPF, [1024, 5]],  # 9
  ]

# YOLOv5 v6.0 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, C3, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, C3, [256, False]],  # 17 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],  # cat head P4
   [-1, 3, C3, [512, False]],  # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5
   [-1, 3, C3, [1024, False]],  # 23 (P5/32-large)

   [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
  ]


#customize iPython writefile so we can write variables
from IPython.core.magic import register_line_cell_magic

@register_line_cell_magic
def writetemplate(line, cell):
    with open(line, 'w') as f:
        f.write(cell.format(**globals()))


%%writetemplate /content/drive/MyDrive/ASL/yolov5/models/custom_yolov5s.yaml
# parameters
nc: {num_classes}  # number of classes
depth_multiple: 0.33  # model depth multiple
width_multiple: 0.50  # layer channel multiple

# anchors
anchors:
```

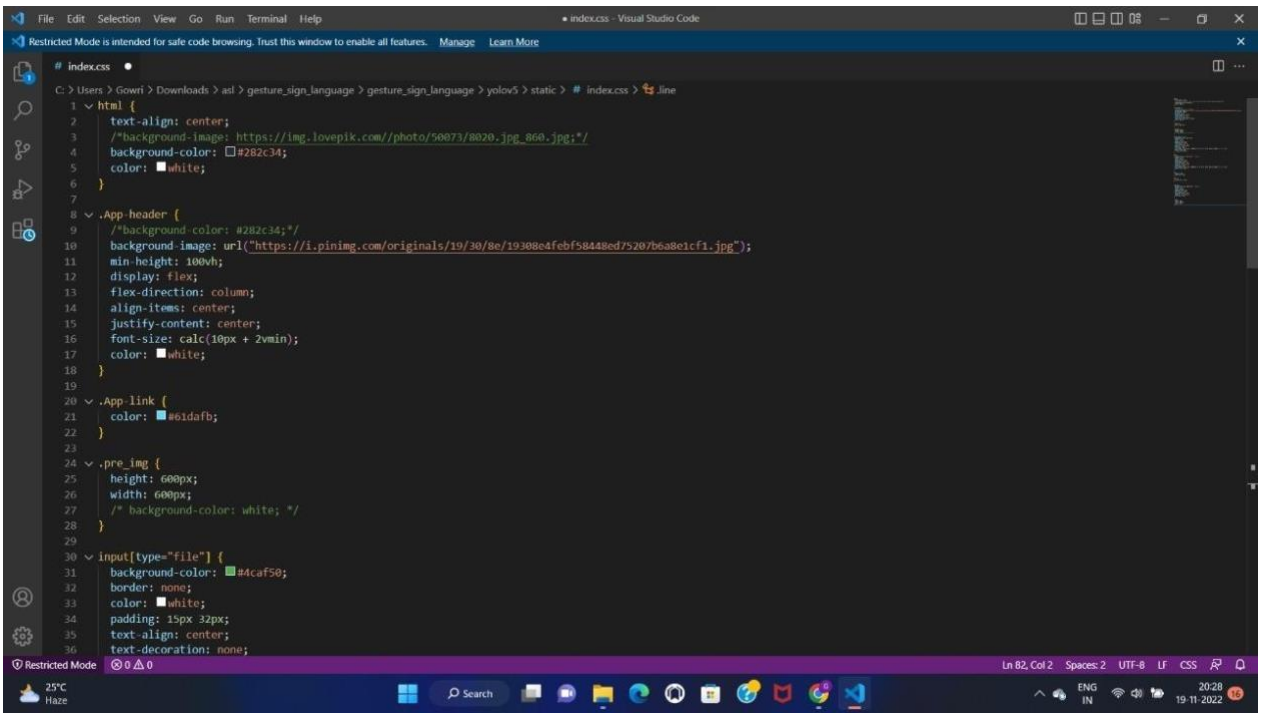
```
- [30,61, 62,45, 59,119]  # P4/16
- [116,90, 156,198, 373,326]  # P5/32


ckbone:
# [from, number, module, args]
[[-1, 1, Focus, [64, 3]],  # 0-P1/2
 [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
 [-1, 3, BottleneckCSP, [128]],
 [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
 [-1, 9, But c Jaae kcst,  [2@] ],
 [-1, 1, com.  lsy2   â,  2 I•]    ¥ »-es.^16
 [-1, 9, BottleneckCSP, [512]],
 [-1, 1, Eat  .   I 8&•  a,  2] ],  r  *-Pfi,'3Z
 [-1, 1, sPP. l1aA.  ls.  9.  1jl l].
 [-1, 3, Be‹ c Jaae‹ kZU',  [ lids ,  Ëzû  e ] | ,  r   0


[[-1,   . Eat .  I fi"• S, '  I•]
 [-1,   .  nn.tanpa.  leo"•  2,  öorøãc " ] ] .
 [[-1, ó], l,  Eon‹ ac ,   l'  I•l      a  ¥zc  bar ba          P•
 [-1, 3, BottleneckCSP, [512, False]],  # 13

 [-1, 1, Conv, [256, 1, 1]],
 [-1, 1, nn.Upsample, [None, 2, 'nearest']],
 [[-1, ‹] , 1.  Eon‹ ac.  l'  I•l      a    ac  bar koxia  PB
 [-1, 5.  Be   c Jaae   kŁU',  [ 2  6,   I aJ ¥+] ],   r  IN  ¢R5.^8 •       JI }

 |-*.  1, Conv, [256, 3, 2]],
 [[-1, 14], 1, Concat, [1]],  # cat head P4
 [-1, 3, BottleneckCSP, [512, False]],  # 20 (P4/16-medium)

 |-*.  1, Conv, [512, 3, 2]],
 [[-1, 10], 1, Concat, [1]],  # cat head P5
 [-1, 3, BottleneckCSP, [1024, False]],  # 23 (P5/32-large)

 [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
```

```
!python train.py --img 416 --batch 16 --epochs 100 --data '../data.yaml' --cfg ./models/cu

     /content/drive/MyDrive/ASL/yolov5
     train: weights=, cfg=./models/custom_yolov5s.yaml, data=../data.yaml, hyp=data/hyp
     Command 'git fetch origin' timed out after 5 seconds
     YOLOv5 🚀 v6.2-226-gfde7758 Python-3.7.15 torch-1.12.1+cu113 CPU

     hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_e
```

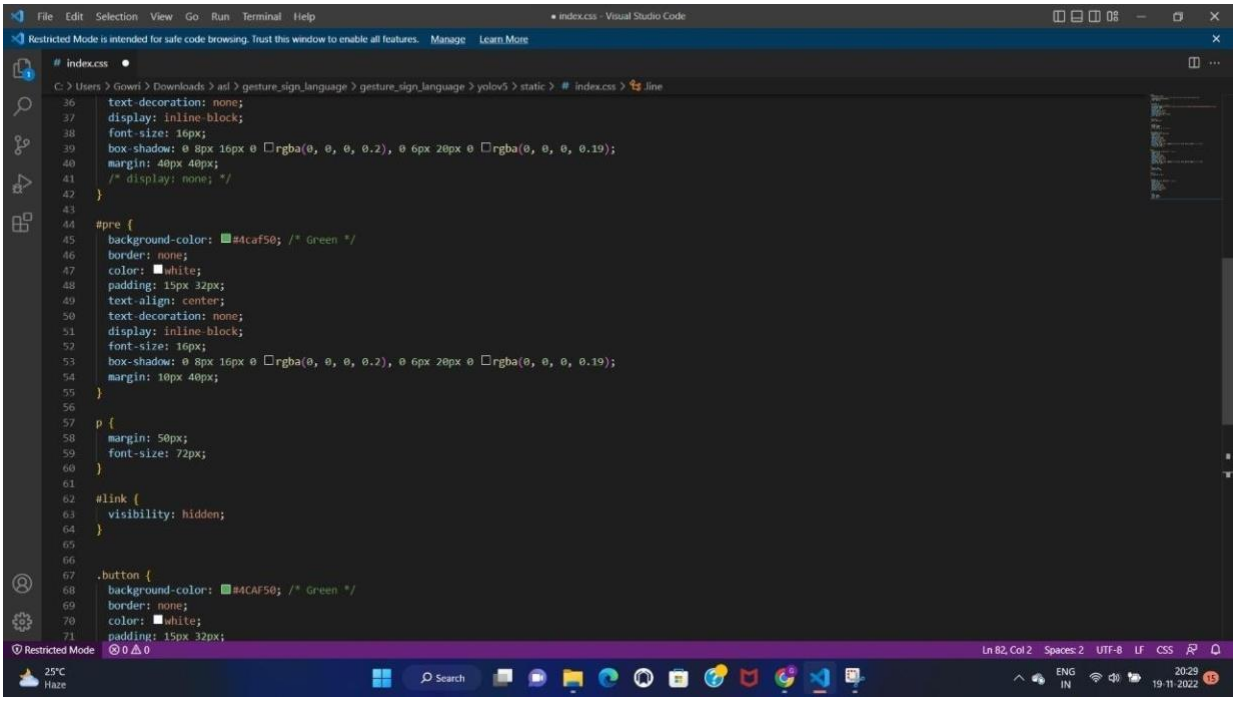


Reference link: http://localhost:8888/notebooks/Downloads/ASL.ipynb

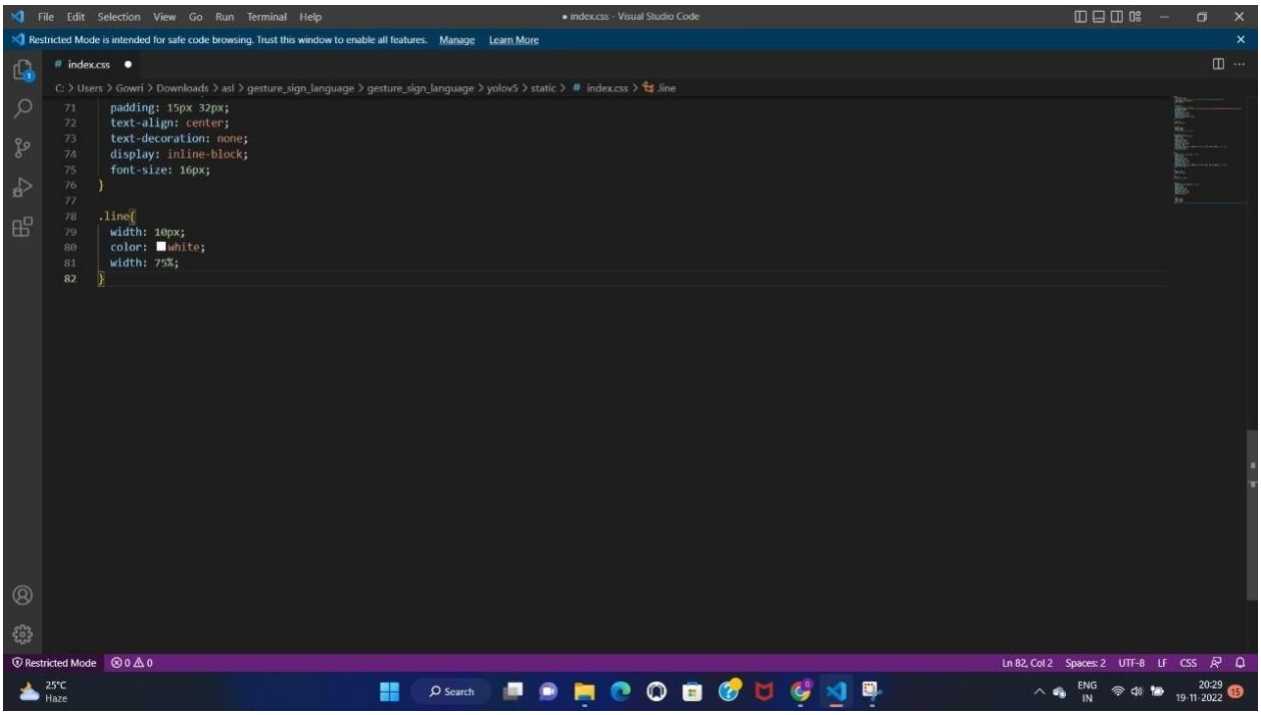
CSS:



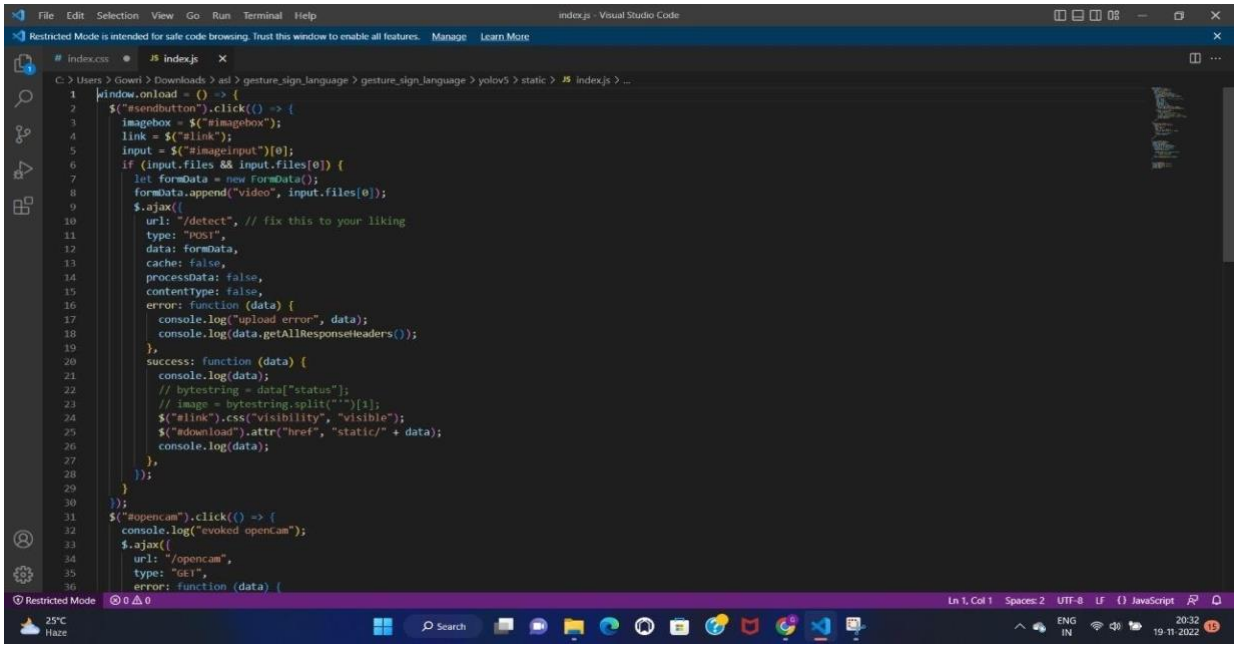
```
html {
  text-align: center;
  /*background-image: https://img.lovepik.com//photo/50073/8020.jpg_860.jpg;*/
  background-color: #282c34;
  color: white;
}

.App-header {
  /*background-color: #282c34;*/
  background-image: url("https://i.pinimg.com/originals/19/30/8e/19308e4febf58448ed75207b6a8e1cf1.jpg");
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

.pre_img {
  height: 600px;
  width: 600px;
  /* background-color: white; */
}

input[type="file"] {
  background-color: #4caf50;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
```

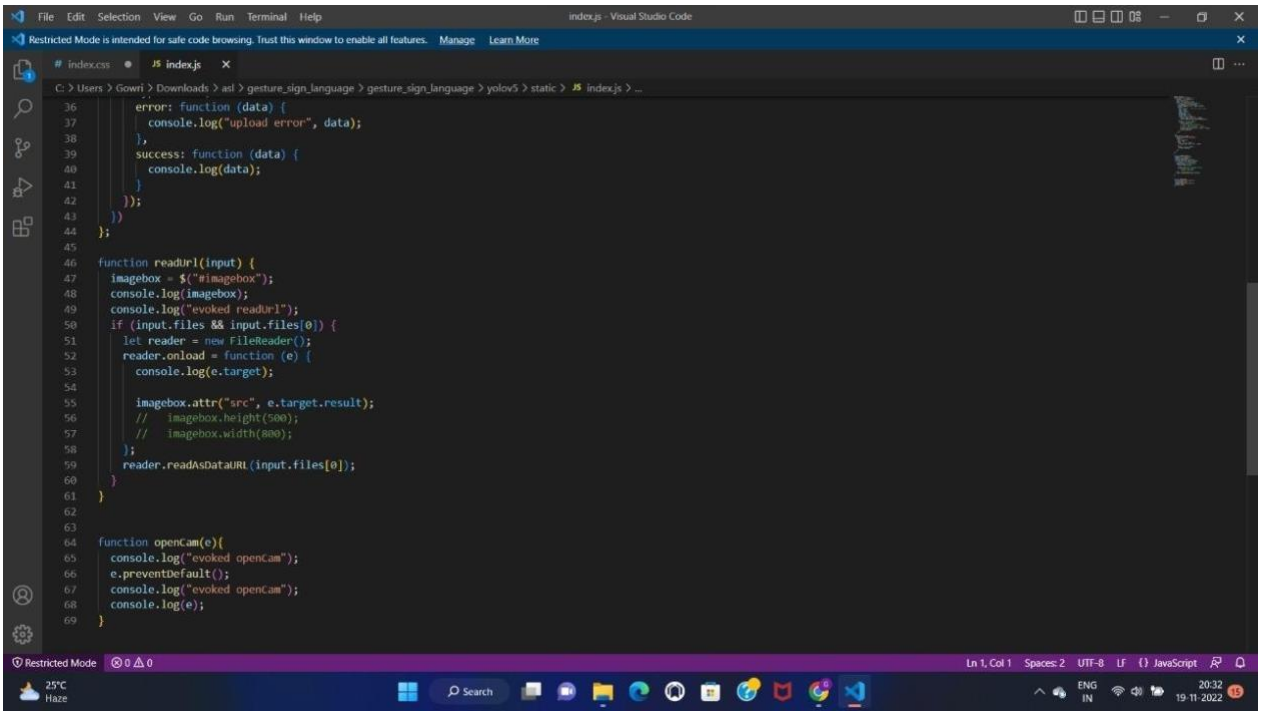


```
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  box-shadow: 0 8px 16px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
  margin: 40px 40px;
  /* display: none; */
}

#pre {
  background-color: #4caf50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  box-shadow: 0 8px 16px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
  margin: 10px 40px;
}

p {
  margin: 50px;
  font-size: 72px;
}

#link {
  visibility: hidden;
}


.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
```


File Edit Selection View Go Run Terminal Help
• index.css - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
index.css
C: > Users > Gowri > Downloads > asl > gesture_sign_language > gesture_sign_language > yolov5 > static > # index.css > .line
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
}
.line{
width: 10px;
color: white;
width: 75%;
}
Restricted Mode
Ln 82, Col 2 Spaces: 2 UTF-8 LF CSS
25°C Haze
Search
ENG IN
20:29 19-11-2022

JSS:

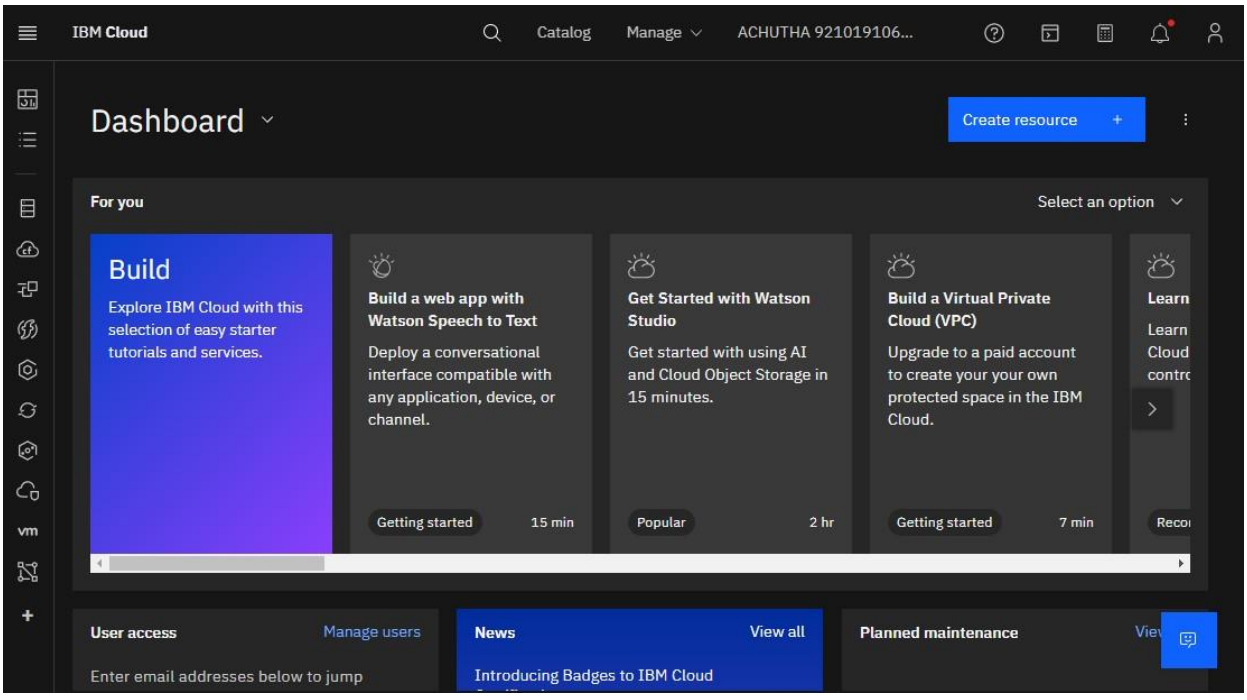

File Edit Selection View Go Run Terminal Help
index.js - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
index.css
index.js
C: > Users > Gowri > Downloads > asl > gesture_sign_language > gesture_sign_language > yolov5 > static > index.js > ...
window.onload = () => {
$("#sendbutton").click(() => {
imagebox = $("#imagebox");
link = $("#link");
input = $("#imageinput")[0];
if (input.files && input.files[0]) {
let formData = new FormData();
formData.append("video", input.files[0]);
$.ajax({
url: "/detect", // fix this to your liking
type: "POST",
data: formData,
cache: false,
processData: false,
contentType: false,
error: function (data) {
console.log("upload error", data);
console.log(data.getAllResponseHeaders());
},
success: function (data) {
console.log(data);
// bytestring = data["status"];
// image = bytestring.split("'")[1];
$("#link").css("visibility", "visible");
$("#download").attr("href", "static/" + data);
console.log(data);
},
});
}
});
$("#opencam").click(() => {
console.log("evoked openCam");
$.ajax({
url: "/opencam",
type: "GET",
error: function (data) {
Restricted Mode
Ln 1, Col 1 Spaces: 2 UTF-8 LF JavaScript
25°C Haze
Search
ENG IN
20:32 19-11-2022



6. TESTING

6.1 Test Cases

6.2 User Acceptance Testing

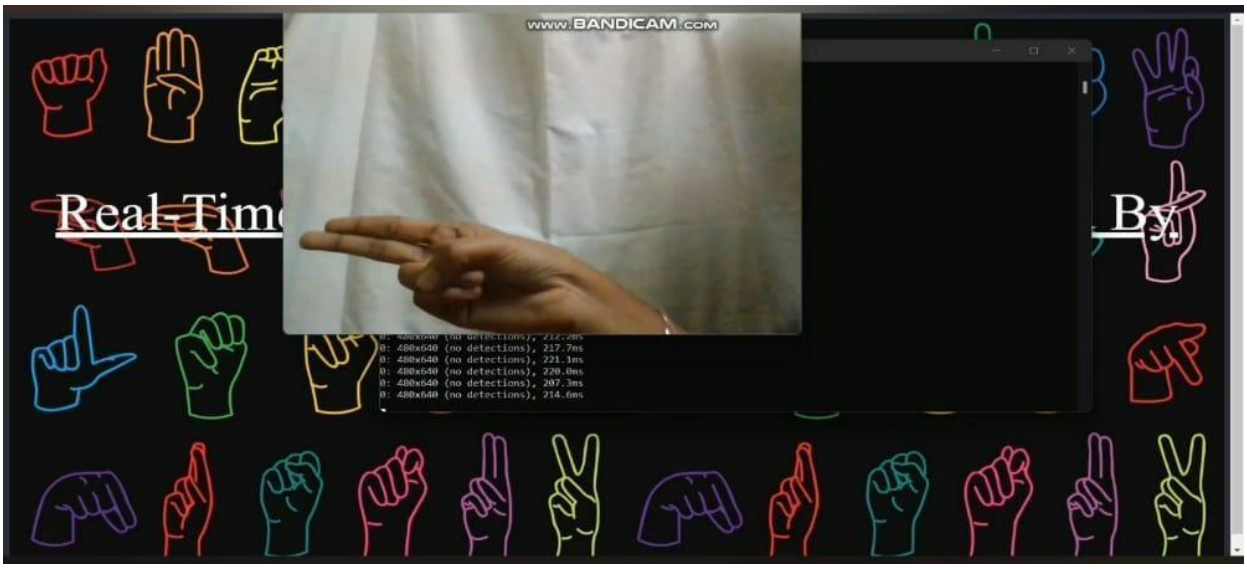


7. RESULTS

7.1 Performance Metrics

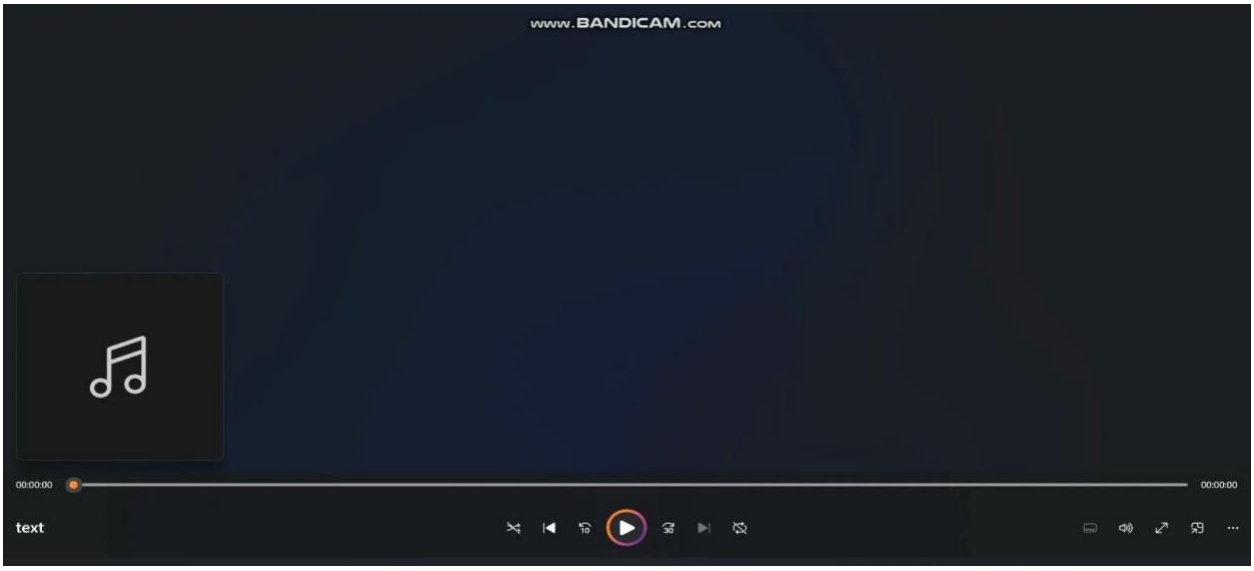
1. Output Screen

2. Camera will pop up



3. It will detect and display in command prompt

4. It will convert detected alphabet to audio



10. ADVANTAGES

- Ease of communication between specially abled and a normal person.
- Easy to handle both specially abled and a normal person.
- Help full in emergency situations.
- Rapid Response.

DISADVANTAGES

- Content delivery need to be improved.
- Improper translation may occur.
- May need high speed data connectivity.

11. CONCLUSION

The main objective of this research has been achieved successfully.

Gesture interpretation works best in case users who understand sign language may interact with people who are unfamiliar with sign language. Speech interpretation is helpful for sign language non-speakers who want the accompanying hand sign to be understood. Room conditions such as lighting can play a role in predicting the outcome of poor lighting. The light that is either too bright or too dim will result in inaccurate hand segmentation, resulting in inaccurate gesture prediction. The type of inaccuracy can emerge from the user's peripherals, such as poor web camera performance or poor microphone quality. In a nutshell, the development of technology is essential, and its deployment in sign language is highly critical. It will serve to bring efficiency in communication, not only to the deaf and dumb but those with the ability to hear and speak as well. In addition to creating opportunities for their career growth, it will enhance their social life through effective communication. Making an impact and changing the lives of the deaf and dump through technology will be an innovation of the year worth the time and resources.

12. FUTURE SCOPE

We can develop a model for ISL word and sentence level recognition. This will require a system that can detect changes with respect to the temporal space. 2. We can develop a complete product that will help the speech and hearing impaired people, and thereby reduce the communication gap.

13. APPENDIX

Source Code

https://github.com/IBM-EPBL/IBM-Project-39684-1660488681

GitHub & Project Demo Link

https://drive.google.com/file/d/1eMYVgzko4dS3-ot2b7Ii3LpS7B9WrAnT/view?usp=share_link