PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYATHIRAN)

PROJECT REPORT

Real-Time Communication System Powered by AI for the Specially Abled

SSN College of Engineering

Team ID: PNT2022TMID52960

Team members:

Riyanka Rajakumar (193002084)

Rohit D (193002085)

Rohith Ram A(193002087)

R Samyuktha (193002091)

Department of Electronics and Communication

CONTENTS

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
8. **TESTING**
   - 8.1 Test Cases
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    - Source Code
    - GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

One of the most underrepresented demographic of people in our country are differently abled people. The technology to assist them is developing fast but no significant products have been developed that will be beneficial to them. Communications between deaf-mute and a normal person has always been a challenging task. It is complicated for mute people to communicate with able people unless they're trained in sign language. Quick communication in emergency situations in especially challenging. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be handy to have a proper conversation between a normal person and an impaired person in any language.

This 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 using 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

One of the most underrepresented demographic of people in our country are differently abled people. The technology to assist them is developing fast but no significant products have been developed that will be beneficial to them. Communications between deaf-mute and a normal person has always been a challenging task. It is complicated for mute people to communicate with able people unless they're trained in sign language. Quick communication in emergency situations in especially challenging. 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 beneficial to have a proper conversation between a normal person and an impaired person in any language.

2. LITERATURE SURVEY

2.1 Existing problem

One of the foundations of human civilization is communication. However, substantial sections of society are not equipped with the ability to communicate easily with others. Millions of people worldwide suffer from varying degrees of hearing loss, speech impairment, or both. Sign Language is a powerful tool to help overcome these disabilities and bridge the gap between hearing and differently-abled people. There are many sign language standards across the world and with the advancement in AI and machine learning, many sign language recognition models have been deployed for a vision-based approach to detect and translate sign language into text, speech, and other formats.

2.2 References

[1]Indian Sign Language Recognition using Deep Learning

P. Mistry, V. Jotaniya, P. Patel, N. Patel and M. Hasan,

2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), 2021, *IEEE*

This paper provides a method of translation from Indian sign language to English words through deep learning models constructed using Convolutional Models and Sequence Models. The goal is to interpret a sign which is recorded through a video camera and sampled into sequences of images. This sequence is then passed through a deep learning model which passes an output vector with the highest value corresponding to the word.

The dataset is divided into training and testing purposes. The training dataset has seven different

signs (Days of week-Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday) The videos corresponding to these signs were recorded and sampled by extracting the frame sequence and developing around 50 sample sequences per word. Image augmentation was done which performed image translation, rotation, and scale to these images and increased its sample size further to about 1500 samples per word making it a 10500 sample size training dataset. Convolutional Neural Network (CNN) has been used for extracting features from images and Long Short Term Memory (LSTM) for analyzing the sequence of these features. The model converged to the training dataset and achieved an accuracy of 99.98%.

[2]Indian Sign Language Recognition using Deep Learning

K. Bantupalli and Y. Xie

2018 IEEE International Conference on Big Data (Big Data), IEEE

The paper presents a vision-based application that offers sign language translation to text thus aiding communication between signers and non-signers. The proposed model takes video sequences and extracts temporal and spatial features from them. Inception, a CNN (Convolutional Neural Network) model, is used for recognizing spatial features. An RNN (Recurrent Neural Network) is then used to train on temporal features. The gesture segments identified and processed by the CNN are classified by LSTM into one of the gesture classes using sequence data.

The dataset used is the American Sign Language Dataset. Each sign was performed five times by a single signer in varying lighting conditions and speed of signing. Each video was broken down by frame to images and then augmented to increase the data set for each sign to 2400 images. The CNN and RNN models were trained independently. The models yielded 90-93% accuracy depending upon the specifics of the architecture.

One of the problems the model faced is with facial features and skin tones. The model also suffered from loss of accuracy with the inclusion of faces, as faces of signers vary and performed

poorly when there was variation in clothing.

[3]Bengali Sign Language Recognition Using Deep Convolutional Neural Network

M. A. Hossen, A. Govindaiah, S. Sultana and A. Bhuiyan

2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), IEEE

In this paper, a new method for Bengali Sign Language Recognition using Deep Convolutional Neural Networks (DCNN) is proposed. The method is built to recognize static hand signs of 37 letters of the Bengali alphabet. The dataset comprises 37 hand signs (total 1147 images), in which each sign class includes 31 images taken at varying distances, orientation and intensity levels. The images represent the signs of the Bengali Language Alphabet as illustrated in the Bengali Sign Language Dictionary.

The images were augmented through many random transformations. Using the concept of transfer learning and fine-tuning, the VGG16 network was pre-trained. The pre-trained network is run on the dataset one time to extract the features necessary for the classification. Hence with the use of deep convolutional neural network and utilizing the learned features from a pre-trained network and fine-tuning the top layers of this network, a high overall recognition rate of 96.33% on the training dataset and 84.68% on the validation dataset have been achieved. This work has however used a very small dataset.

2.3 Problem Statement Definition

Problems Faced:

- Only specially abled people are taught sign language and the common person is unaware its working causing a communication gap.
- Under emergency situations, it is even more difficult for specially abled people to get help.

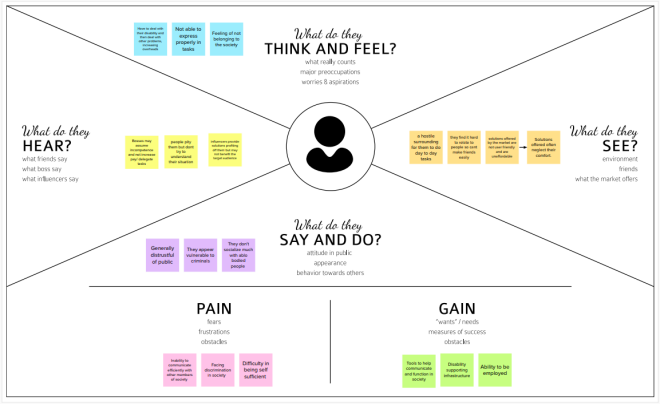
- Non-Emergency normal environments can also be hard for them to navigate needing special assistance.

Our solution:

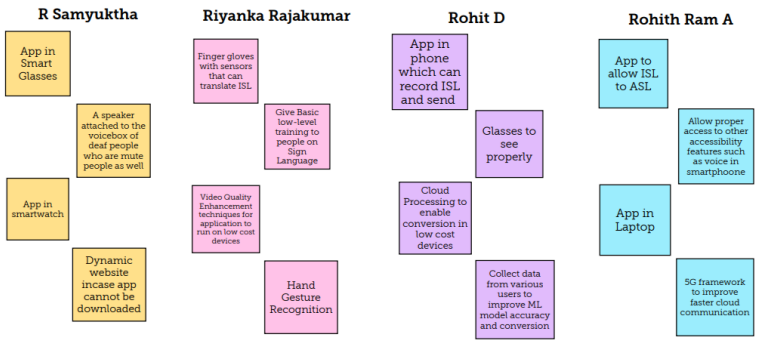
- 1. To build a system which converts sign language to speech and vice versa.
- 2. Make Use of various CNN models such as Inception and RNN to learn the conversion.
- 3. Convert various sign languages such as ASL and ISL.
- 4. Enable multi language conversion such as Tamil and Hindi.
- 5.Host in cloud to enable running on low cost mobile phones with low processing power.

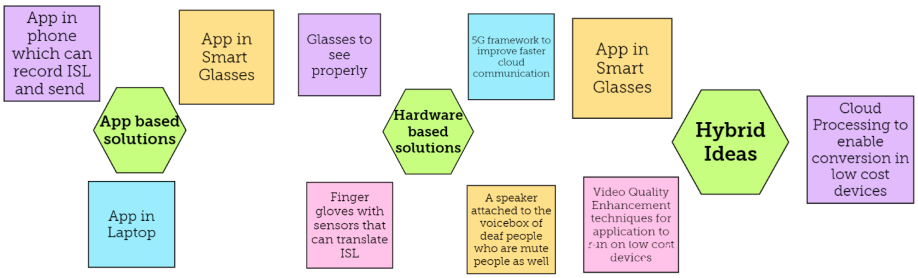
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



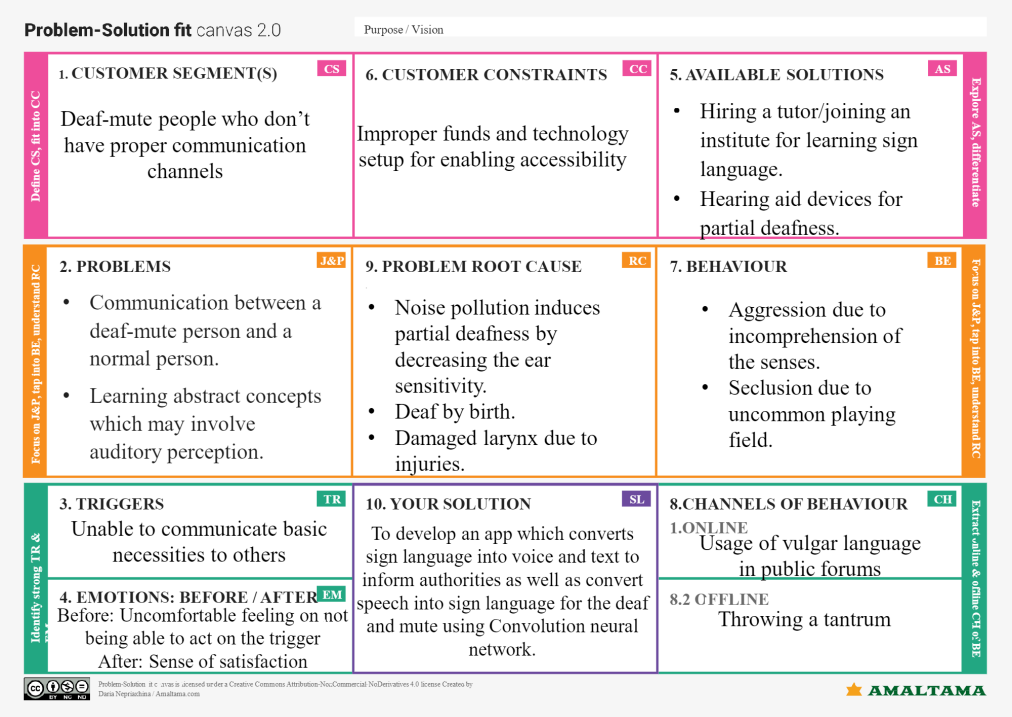


3.3 Proposed Solution

To facilitate communication between a differently abled person and a normal person during emergency situations our solution aims to to develop an application which converts sign language into a human voice in the desired language as well as convert speech into understandable sign language for the deaf and dumb using Convolution neural network. The

model makes use of a variation of CCN model called U-net. The application converts the (ISL) Indian Sign Language to text and sending it to the necessary authorities during emergency situations.

3.4 Problem Solution fit



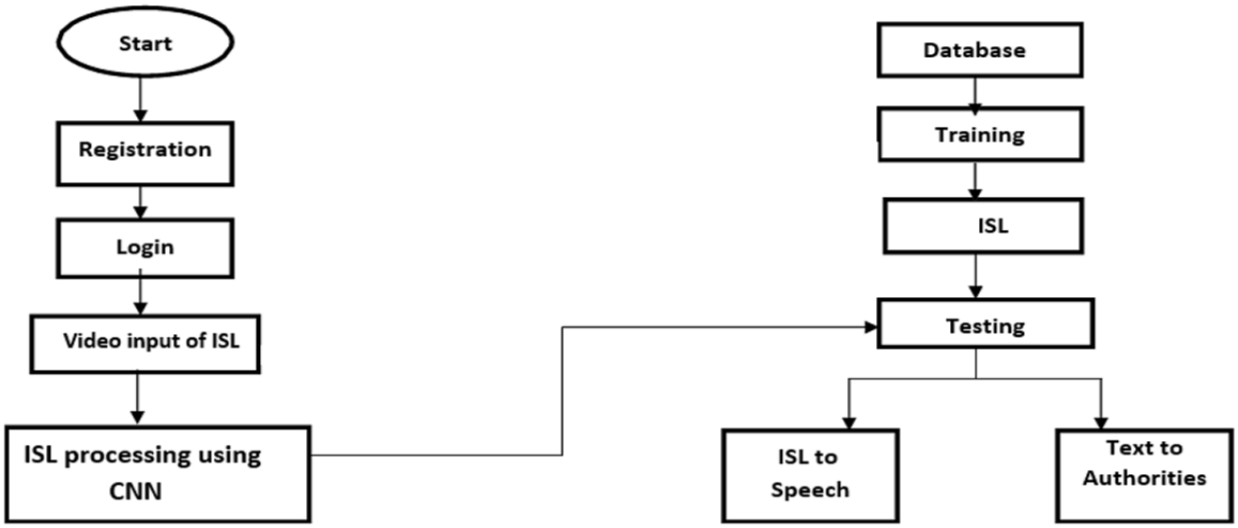
4. REQUIREMENT ANALYSIS

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

User Stories

Use the below template to list all the user stories for the product.

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 mobile number	I can register and access my account	High	Sprint-1
		USN-2	As a user, I can register for the application through my gmail.	I can register and access my account	High	Sprint-1
		USN-3	As a user, I can register for the application through facebook/google account.	I can register and access my account	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register and access my account	Medium	Sprint-2
	Confirmation	USN-5	As a user, I can receive a confirmation via OTP in gmail/SMS	I can receive confirmation via OTP	High	Sprint-1
		USN-6	As a user, I can receive confirmation by username and password.	I can receive confirmation via username and password	Low	Sprint-2
	Login	USN-7	As a user I can auto login using cache		Medium	Sprint-2
		USN-8	As a user I can login using OTP		Low	Sprint-2
		USN-9	As a user I can login using my google account or the account created for the application.		High	Sprint-1
	User preferences	USN-10	As a user I can enable normal/emergency communication depending upon the need which rises		High	Sprint-1
		USN-11	As a user I can choose ASL/iSL conversion		Low	Sprint-2
		USN-12	As a user I can convert sign language to various texts		High	Sprint-1
		USN-13	As a user, I can enable data sharing to improve conversion accuracies.		Low	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

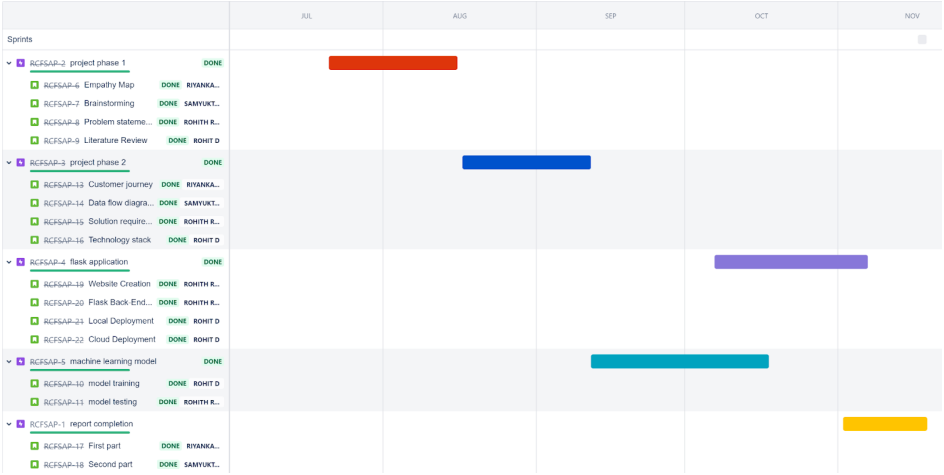
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Datasets and preprocessing	USN-1	Creating and segregating the datasets into testing and training	2	Low	Riyanka Rajakumar
Sprint-1		USN-2	Importing the Imagedatagenerator library and configuring it for image preprocessing	3	Medium	Rohit D
Sprint-1		USN-3	Applying the ImageDataGenerator functionality to train and test the data.	3	Medium	Rohit D
Sprint-2	Model Building (training and testing)	USN-4	Initializing the model	2	Low	Samyuktha R
Sprint-2		USN-5	Adding the convolution layer	3	Medium	Rohith Ram A
Sprint-2		USN-6	Adding the pooling layer	3	Medium	Rohit D
Sprint-2		USN-7	Adding the flatten layer	3	Medium	Rohith Ram A
Sprint-2		USN-8	Adding the dense layer	3	Medium	Riyanka Rajakumar
Sprint-2		USN-9	Compiling the model	2	Low	Samyuktha R
Sprint-2		USN-10	Fit and saving the model	2	Low	Riyanka Rajakumar
Sprint-2		USN-11	Testing the model	3	Medium	Rohith Ram A

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Flask application	USN-12	Building the HTML page	5	Very High	Samyuktha R
Sprint-3		USN-13	Running the flask application	5	Very High	Rohith Ram A
Sprint-4	Deploying the model on IBM Cloud	USN-14	Register for IBM Cloud	2	Low	Riyanka Rajakumar
Sprint-4		USN-15	Training the image classification model	4	High	Samyuktha R
Sprint -4		USN-16	Deployment of application on IBM Cloud	5	Very High	Rohit D

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	8	4 Days	22 Oct 2022	26 Oct 2022	20	26 Oct 2022
Sprint-2	21	6 Days	27 Oct 2022	02 Nov 2022	21	02 Nov 2022
Sprint-3	10	6 Days	04 Nov 2022	10 Nov 2022	10	10 Nov 2022
Sprint-4	11	5 Days	12 Nov 2022	17 Nov 2022	11	17 Nov 2022

6.3 Reports from JIRA



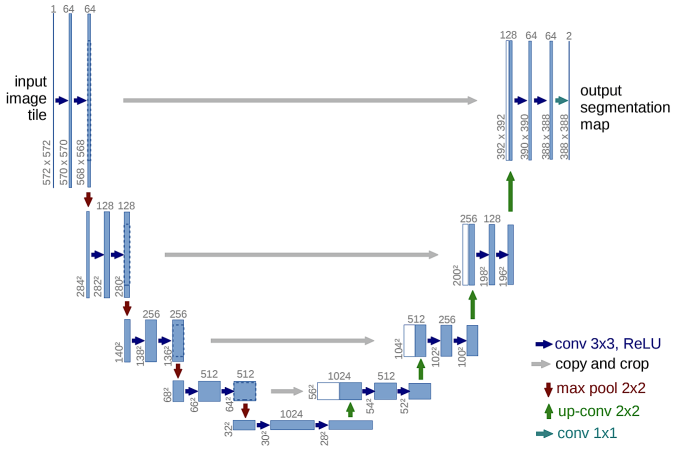
7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Image Segmentation is the process by which a digital image is partitioned into various subgroups (of pixels) called Image Objects, which can reduce the complexity of the image, and thus analysing the image becomes simpler.Assignment of labels to pixels is done and the pixels with the same label fall under a category where they have some or the other thing common in them.
Image segmentation performs localisation and classification.Localization means finding the location (pixels) of a particular object within a much larger image. Classification means to classify the object that has been localized within the image.

U-Net is a convolutional neural network that was developed for image segmentation. The network is based on a fully convolutional network whose architecture was modified and extended to work with fewer training images and yield fast and precise segmentation.
Architecture:
U-net architecture is symmetric and consists of two major parts:

- The left part is called the contracting path which classifies the object, constituted by the general convolutional process.
- The right part is an expansive path which localizes the object, constituted by transposed 2D convolutional layers.



Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

The image is input at the beginning of the network (left-top). The data is then propagated through all possible paths and, in the end, the segmentation map comes out.The main idea is to supplement a usual contracting network by successive layers, where upsampling operators replace pooling operations.

The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling.

At each downsampling step the number of feature channels are doubled.

Every step in the expansive path consists of an up-sampling of the feature map followed by a 2x2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU.

The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes.

In total the network has 23 convolutional layers.

Hence these layers increase the resolution of the output.

Features of U-net:

- U-Net learns segmentation in an end-to-end setting.

  You input a raw image and get a segmentation map as the output.
- U-Net is able to localize precisely and distinguish borders.

  Performs classification on every pixel so that the input and output share the same size.
- U-Net uses limited annotated images.

  Data augmentation with elastic deformations reduces the number of annotated images required for training.

7.2 Feature 2

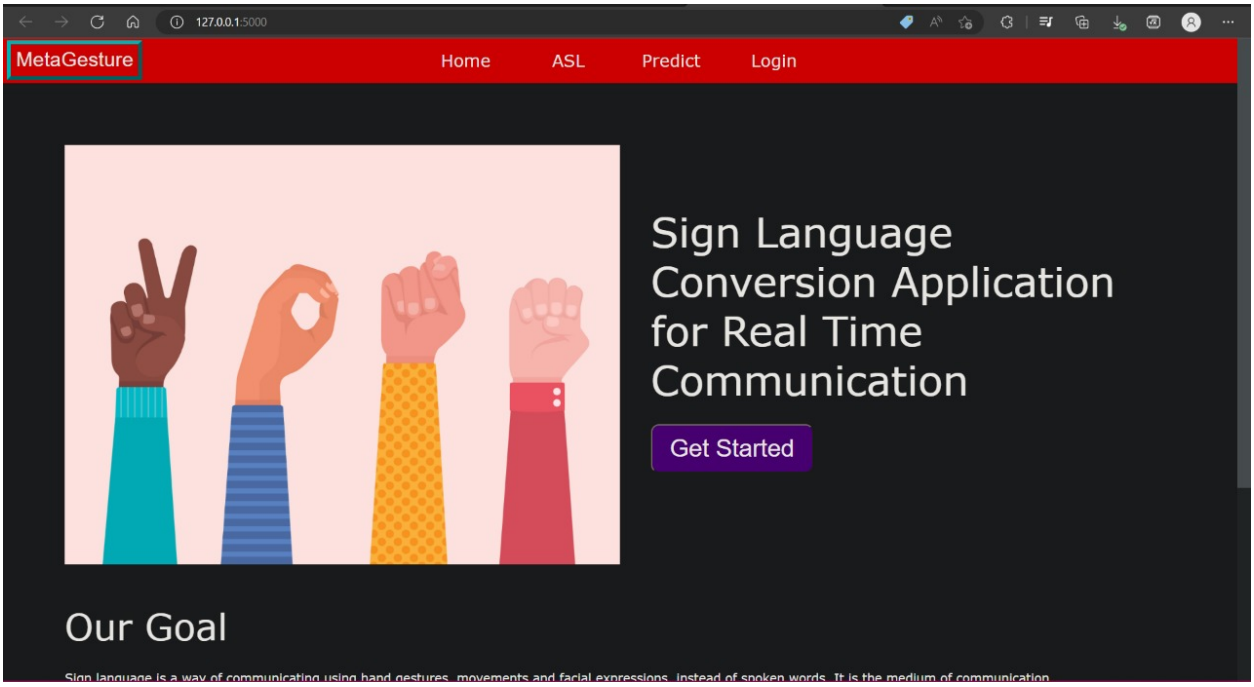
The model contains a FLASK Application completed with login and registration forms to allow the specially abled persons to have an overall experience. Using frameworks like OpenCV and Keras it was possible to include real time prediction of sign language using the builtin camera of the device the website is running on.

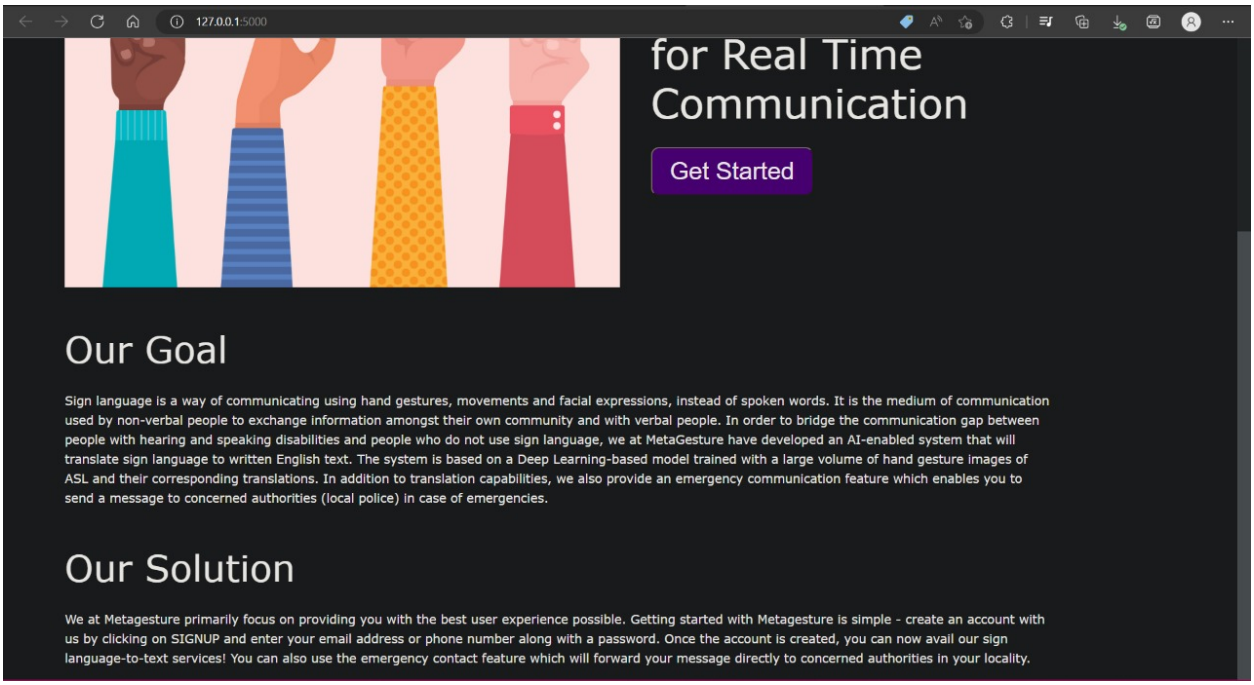
The signing facility allows for personalization of the model and helps improves the overall experience, but the prediction facility can be used without the need to sign in. This improves privacy of the users

The main features of the application are listed below:

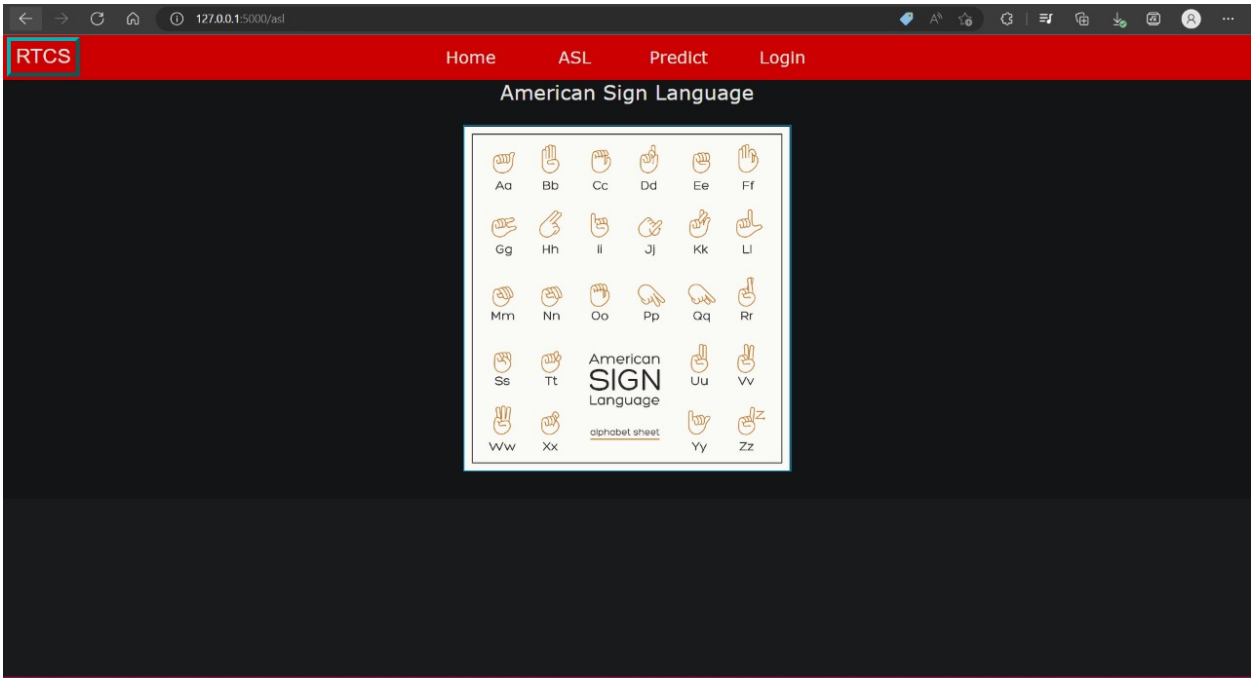
- Real time prediction of sign language and natural interpretation.
- An account management system.
- Supports American Sign Language.
- Works across all platforms and devices which support web browsing.
- No signing required - predictions.

Home page:

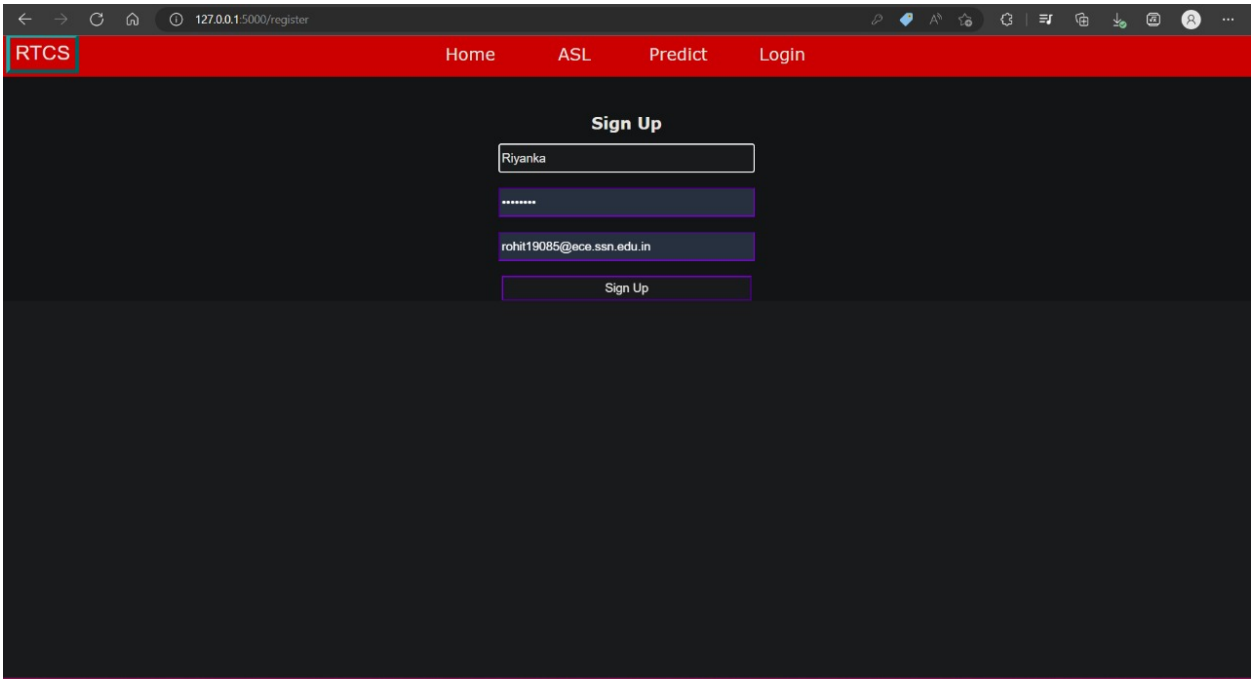




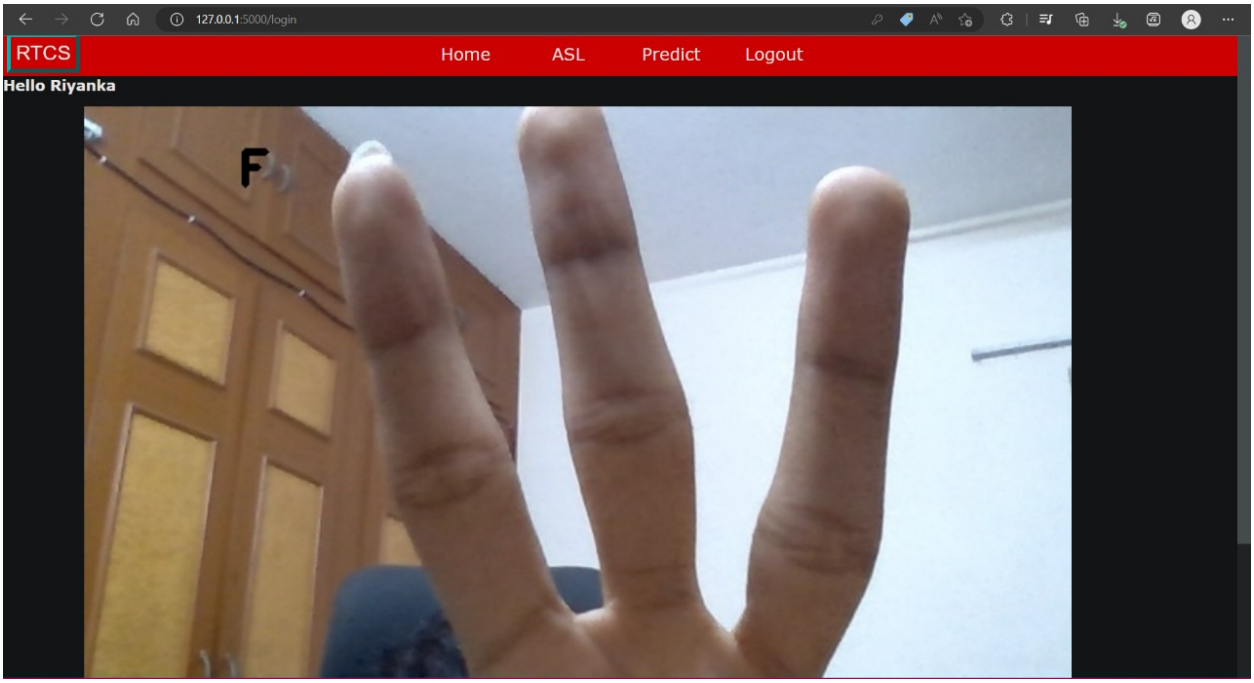
ASL Sign language page:



Register page:



Prediction page:



8. TESTING

8.1 **Testing the Model**

```
In [41]: path="C:/Users/riyan/Downloads/conversation engine for deaf and dumb (1)/Dataset/test_set/H/20.png"
         img=load_img(path,target_size=(100,100))
         img
```

Out[41]:

```
In [35]: arr= img_to_array(img)
         frame=cv2.imread(path)
         data=detect(frame)

         1/1 [==============================] - 0s 180ms/step
         THE PREDICTED LETTER IS  H
```

```
In [42]: path="C:/Users/riyan/Downloads/conversation engine for deaf and dumb (1)/Dataset/test_set/B/2.png"
         img=load_img(path,target_size=(100,100))
         img
```

Out[42]:

```
In [39]: rr= img_to_array(img)
         frame=cv2.imread(path)
         data=detect(frame)

         1/1 [==============================] - 0s 21ms/step
         THE PREDICTED LETTER IS  B
```

8.2 **Test Cases**

S. No.	Test Scenarios
1.	Verify if user can see the login popup whenever he/she clicks on the login button.
2.	Verify if user is able to log into the application with invalid credentials.
3.	Verify if the user is able to see video feed in predict page
4.	Verify if the user is able to see predicted alphabet generated.
5.	Verify if the user can login again after logging out of the application successfully.
6.	Verify the UI in the register page of the application.
7.	Verify user information is stored in database for further communications.
8.	Verify the UI elements in the register page of the application.

8.3 **User Acceptance Testing**

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	4	2	3	19
Duplicate	0	0	3	0	3
External	2	3	0	1	6
Fixed	9	2	4	20	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	20	14	13	26	74

9. RESULTS

9.1 Performance Metrics

```
Epoch 18/25
79/79 [==============================] - 55s 698ms/step - loss: 6.0743e-04 - accuracy: 0.9998 - val_loss: 0.1199 - val_accur
acy: 0.9840
Epoch 19/25
79/79 [==============================] - 56s 709ms/step - loss: 9.5635e-05 - accuracy: 1.0000 - val_loss: 0.0986 - val_accur
acy: 0.9840
Epoch 20/25
79/79 [==============================] - 56s 701ms/step - loss: 5.7821e-04 - accuracy: 0.9999 - val_loss: 0.1204 - val_accur
acy: 0.9844
Epoch 21/25
79/79 [==============================] - 58s 735ms/step - loss: 0.0041 - accuracy: 0.9986 - val_loss: 0.0084 - val_accuracy:
0.9987
Epoch 22/25
79/79 [==============================] - 54s 685ms/step - loss: 0.0082 - accuracy: 0.9973 - val_loss: 0.0501 - val_accuracy:
0.9849
Epoch 23/25
79/79 [==============================] - 55s 688ms/step - loss: 0.0041 - accuracy: 0.9985 - val_loss: 0.0147 - val_accuracy:
0.9964
Epoch 24/25
79/79 [==============================] - 56s 710ms/step - loss: 0.0021 - accuracy: 0.9994 - val_loss: 0.0491 - val_accuracy:
0.9853
Epoch 25/25
79/79 [==============================] - 56s 710ms/step - loss: 0.0017 - accuracy: 0.9995 - val_loss: 0.0358 - val_accuracy:
0.9929
```

10. ADVANTAGES & DISADVANTAGES

Advantages:

- Easy to use.
- Fills the communication gap between the specially abled person and a normal person.
- Useful in emergency situations

Disadvantages:

- Will take a while to get accustomed in navigating the application.
- Will need active and proper internet connection

11. CONCLUSION

We have developed a system that converts the sign language into text in the desired language to convey a message to normal people, as well as text to understandable sign language for the deaf and dumb. We have made use of a variation of convolution neural network model called U-net 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 texts which can be sent to the authorities to get timely help.

12. FUTURE SCOPE

Since we have used ASL in this project we can further extend this project using Indian Sign languages also.Conversion of sign language to all the regional languages of India can be implemented. We can introduce communities in the app and help deaf-mute people from across the world to interact and learn other sign languages from each other. Emergency contact provision can be extended to be able to contact local ambulance, fire station etc.

13. APPENDIX

Source Code

App.py

```
from flask import Flask, Response, render_template
from camera import Video
import cv2
from keras.utils import load_img,img_to_array
import numpy as np
```

```
from skimage.transform import resize
from keras.models import load_model
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mysqldb import MySQL
import MySQLdb.cursors
import re

app.secret_key = 'your secret key'

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '12345678'
app.config['MYSQL_DB'] = 'geeklogin'

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

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

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

@app.route('/login', methods =['GET', 'POST'])
def login():
```

```
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM accounts WHERE username = % s AND password = %
s', (username, password, ))
        account = cursor.fetchone()
        if account:
            session['loggedin'] = True
            session['id'] = account['id']
            session['username'] = account['username']
            msg = 'Logged in successfully !'
            return render_template('index.html', msg = msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect(url_for('login'))

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form
```

```
and 'email' in request.form :
        username = request.form['username']
        password = request.form['password']
        email = request.form['email']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM accounts WHERE username = % s', (username, ))
        account = cursor.fetchone()
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'Username must contain only characters and numbers !'
        elif not username or not password or not email:
            msg = 'Please fill out the form !'
        else:
            cursor.execute('INSERT INTO accounts VALUES (NULL, % s, % s, % s)', (username,
password, email, ))
            mysql.connection.commit()
            msg = 'You have successfully registered !'
            return render_template('login.html', msg = msg)
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)

model=load_model('C:/Users/rohit/OneDrive/Desktop/VII/IBM-Project-19465-
1659698319/Project Development Phase/Sprint 2/asl.h5')
def gen(video):
    while True:
```

```
        success, image = video.read()

        copy=image.copy()
        cv2.imwrite('img.jpg',copy)
        copy_img=load_img('img.jpg',target_size=(64,64,1))
        img=img_to_array(copy_img)
        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(op[pred])
        cv2.putText(image,op[pred],(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)

        ret, jpeg = cv2.imencode('.jpg', image)
        frame = jpeg.tobytes()
        yield (b'--frame\r\n'b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

video = cv2.VideoCapture(0)
@app.route('/video_feed')
def video_feed():
    global video
    return Response(gen(video),mimetype='multipart/x-mixed-replace; boundary=frame')

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

Camera.py

```
import cv2
```

```
import numpy as np
from keras.models import load_model
from keras.preprocessing import image
import os
from keras.utils import load_img,img_to_array

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('C:/Users/rohit/OneDrive/Desktop/VII/IBM-Project-19465-1659698319/Project Development Phase/Sprint 2/asl.h5') # Execute Local Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None

    def __del__(self):
        k = cv2.waitKey(1)
        self.video.release()

    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame,(640,480))
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

Home.html

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>MetaGesture</title>
    <meta charset="utf-8">
    <meta name= "viewport" content="initial-scale=1" width="device-width">
    <link rel="stylesheet" type="text/css" href="/static/base.css">
    <link rel="icon" href="/images/logo.png">
    <meta property="og:image" src="/static/logo.png">
    <meta property="og:image:type" content="image/png">
    <meta property="og:image:width" content="640">
    <meta property="og:image:height" content="480">
    <meta property="og:type" content="website" />
    <meta property="og:title" content="MetaGesture">
     <meta property="og:description" content="Get Stats for your Youtube Music and Share with your friends">
    <meta name="keywords" content=" ASL,ISL,Sign,Language,Communication">
    <meta name="author" content="Rohit D">
</head>
<body>
    <div class="top col-lg-12 col-md-12 col-sd-12">
        <div class="col-lg-4 col-md-3 col-sd-12">
            <div class="theplaylist">
            <button class="butnoob"><a class="butnooblink" href="/">RTCS</a></button>
            </div>
        </div>
        <div class="navbut col-lg-4 col-md-1 col-sd-3">
            <div class="navbuts navbuts1">
                <a class="navname"></a>
            </div>
        </div>
```

```
    <div class="navbut col-lg-1 col-md-2 col-sd-3">
        <div class="navbuts navbuts1">
        <a class="navname" href="/">Home</a>
        </div>
    </div>
    <div class="navbut col-lg-1 col-md-2 col-sd-2">
        <div class="navbuts">
        <a class="navname" href="/asl">ASL</a>
        </div>
    </div>
    <div class="navbut col-lg-1 col-md-2 col-sd-2">
        <div class="navbuts">
        <a class="navname" href="/predict">Predict</a>
        </div>
    </div>
    <div class="navbut col-lg-1 col-md-2 col-sd-2">
        <div class="navbuts">
        <a class="navname" href="/login">Login</a>
        </div>
    </div>
</div>
<div class="mainbody">
    <link rel="stylesheet" type="text/css" href="/static/home.css">
    <div class="infoall">
    <div class="col-lg-6 col-md-6 col-sd-12">
        <img class="metaimg" src="/static/index.png" >
    </div>
    <div class="col-lg-6 col-md-6 col-sd-12">
                <p class="metainfo">Sign Language Conversion Application for Real Time
```

```
Communication</p>
            <button class="metabutton"><a style="text-decoration:none;color:white;height:50px;" href="login.html">Get Started</a></button>
        </div>
        <div class="col-lg-12 col-md-12 col-sd-12">
           <p class="metahow"> How it Works </p>
           <p class="metah">Introduction</p>
              <p class="metasub">Wanted to create a playlist and find stats for your YouTube and YouTube Music Account? This website is made just for you. The Playlist is an interactive website which automatically creates and stores
              your youtube playlists, which can be shared with friends.</p>
           <p class="metah">Home</p>
           <p class="metasub">Creates playlists on a month-by-month basis</p>
             <p class="metasub">The Monthly Playlist contains the list of songs and the number of times listened to it in the given month</p>
           <p class="metah">Songs</p>
           <p class="metasub">View Number of times you listened to your favouorite songs</p>
           <p class="metah">Artists</p>
           <p class="metasub">Same as Songs tab, but for your favourite artists</p>
           <p class="metah">Update</p>
            <p class="metasub">This is the heart of the website. To create the playlist, your Youtube and Youtube
              Music Watch-History is needed. Follow the Instructions in the update tab upload your
               google data here so that the playlists can be created. You can Update your playlist how many ever
              times you want and as you please.</p>
        </div>
    </div>
    </div>
```

```
    <div id="footer">
            <a target="_blank" class="linkedinname" href="https://www.linkedin.com/in/rohit-d-898384211/">
        Designed by Rohit D <img class="linkedin" src="/static/linkedin.png">
        </a>
    </div>
</body>
</html>
```

Login.html

```
<!DOCTYPE html>
<html>
<head>
    <title>MetaGesture</title>
    <meta charset="utf-8">
    <meta name= "viewport" content="initial-scale=1" width="device-width">
    <link rel="stylesheet" type="text/css" href="/static/base.css">
    <link rel="icon" href="/images/logo.png">
    <meta property="og:image" src="/static/logo.png">
    <meta property="og:image:type" content="image/png">
    <meta property="og:image:width" content="640">
    <meta property="og:image:height" content="480">
    <meta property="og:type" content="website" />
    <meta property="og:title" content="MetaGesture">
     <meta property="og:description" content="Get Stats for your Youtube Music and Share with your friends">
    <meta name="keywords" content=" ASL,ISL,Sign,Language,Communication">
    <meta name="author" content="Rohit D">
```

```
</head>
<body>
   <div class="top col-lg-12 col-md-12 col-sd-12">
      <div class="col-lg-4 col-md-3 col-sd-12">
         <div class="theplaylist">
         <button class="butnoob"><a class="butnooblink" href="/">RTCS</a></button>
         </div>
      </div>
      <div class="navbut col-lg-4 col-md-1 col-sd-3">
         <div class="navbuts navbuts1">
            <a class="navname"></a>
         </div>
      </div>
      <div class="navbut col-lg-1 col-md-2 col-sd-3">
         <div class="navbuts navbuts1">
         <a class="navname" href="/">Home</a>
         </div>
      </div>
      <div class="navbut col-lg-1 col-md-2 col-sd-2">
         <div class="navbuts">
         <a class="navname" href="/asl">ASL</a>
         </div>
      </div>
      <div class="navbut col-lg-1 col-md-2 col-sd-2">
         <div class="navbuts">
         <a class="navname" href="/predict">Predict</a>
         </div>
      </div>
      <div class="navbut col-lg-1 col-md-2 col-sd-2">
```

```
        <div class="navbuts">
        <a class="navname" href="/login">Login</a>
        </div>
      </div>
   </div>
   <div class="mainbody">
      <link rel="stylesheet" type="text/css" href="/static/login.css">
      <div class="body">
         <div class="usernoob"> Sign In to Get Started</div>
         <form action="{{ url_for('login') }}" method="post">
             <div class="msg">{{ msg }}</div>
                    <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/></br></br>
                 <input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox"/></br></br></br>
                <input type="submit" class="btn" value="Sign In"></br></br>
         </form>
         <p class="signup">Don't have an account? Create one
            <a  href="/register">here</a>
         </p>
      </div>
  </div>
   <div id="footer">
            <a  target="_blank"  class="linkedinname"  href="https://www.linkedin.com/in/rohit-d-898384211/">
      Designed by Rohit D <img class="linkedin" src="/static/linkedin.png">
      </a>
   </div>
</body>
```

```
</html>
```

predict.html

```
<!DOCTYPE html>
<html>
<head>
    <title>MetaGesture</title>
    <meta charset="utf-8">
    <meta name= "viewport" content="initial-scale=1" width="device-width">
    <link rel="stylesheet" type="text/css" href="/static/base.css">
    <link rel="icon" href="/images/logo.png">
    <meta property="og:image" src="/static/logo.png">
    <meta property="og:image:type" content="image/png">
    <meta property="og:image:width" content="640">
    <meta property="og:image:height" content="480">
    <meta property="og:type" content="website" />
    <meta property="og:title" content="MetaGesture">
     <meta property="og:description" content="Get Stats for your Youtube Music and Share with your friends">
    <meta name="keywords" content=" ASL,ISL,Sign,Language,Communication">
    <meta name="author" content="Rohit D">
</head>
<body>
    <div class="top col-lg-12 col-md-12 col-sd-12">
        <div class="col-lg-4 col-md-3 col-sd-12">
            <div class="theplaylist">
            <button class="butnoob"><a class="butnooblink" href="/">RTCS</a></button>
            </div>
        </div>
```

```
    <div class="navbut col-lg-4 col-md-1 col-sd-3">
        <div class="navbuts navbuts1">
            <a class="navname"></a>
        </div>
    </div>
    <div class="navbut col-lg-1 col-md-2 col-sd-3">
        <div class="navbuts navbuts1">
        <a class="navname" href="/">Home</a>
        </div>
    </div>
    <div class="navbut col-lg-1 col-md-2 col-sd-2">
        <div class="navbuts">
        <a class="navname" href="/asl">ASL</a>
        </div>
    </div>
    <div class="navbut col-lg-1 col-md-2 col-sd-2">
        <div class="navbuts">
        <a class="navname" href="/predict">Predict</a>
        </div>
    </div>
    <div class="navbut col-lg-1 col-md-2 col-sd-2">
        <div class="navbuts">
        <a class="navname" href="/login">Login</a>
        </div>
    </div>
</div>
<div class="mainbody">
    <link rel="stylesheet" type="text/css" href="static/predict.css">
        <img src="{{ url_for('video_feed') }}" style=" margin-left:100px; width: 80%;height:
```

```
80%;color: rgb(255,255,255);text-align: center;font-size: 20px;"
        alt="Camera Access Not Provided!">
        {% if session.loggedin %}
        <h2>Hi {{session.username}}</h2>
        {% else %}
        <h2>Login to get better experience and help improve website</h2>
        {% endif %}
    </div>
    <div id="footer">
            <a target="_blank" class="linkedinname" href="https://www.linkedin.com/in/rohit-d-
898384211/">
        Designed by Rohit D <img class="linkedin" src="/static/linkedin.png">
        </a>
    </div>
</body>
</html>
```

Register.html

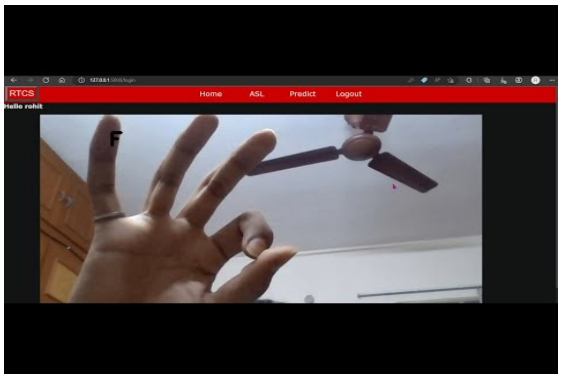
```
<!DOCTYPE html>
<html>
<head>
    <title>MetaGesture</title>
    <meta charset="utf-8">
    <meta name= "viewport" content="initial-scale=1" width="device-width">
    <link rel="stylesheet" type="text/css" href="/static/base.css">
    <link rel="icon" href="/images/logo.png">
    <meta property="og:image" src="/static/logo.png">
    <meta property="og:image:type" content="image/png">
    <meta property="og:image:width" content="640">
```

```
    <meta property="og:image:height" content="480">
    <meta property="og:type" content="website" />
    <meta property="og:title" content="MetaGesture">
     <meta property="og:description" content="Get Stats for your Youtube Music and Share with your friends">
    <meta name="keywords" content=" ASL,ISL,Sign,Language,Communication">
    <meta name="author" content="Rohit D">

</head>
<body>
    <div class="top col-lg-12 col-md-12 col-sd-12">
        <div class="col-lg-4 col-md-3 col-sd-12">
            <div class="theplaylist">
            <button class="butnoob"><a class="butnooblink" href="/">RTCS</a></button>
            </div>
        </div>
        <div class="navbut col-lg-4 col-md-1 col-sd-3">
            <div class="navbuts navbuts1">
                <a class="navname"></a>
            </div>
        </div>
        <div class="navbut col-lg-1 col-md-2 col-sd-3">
            <div class="navbuts navbuts1">
            <a class="navname" href="/">Home</a>
            </div>
        </div>
        <div class="navbut col-lg-1 col-md-2 col-sd-2">
            <div class="navbuts">
            <a class="navname" href="/asl">ASL</a>
```

```
        </div>
      </div>
      <div class="navbut col-lg-1 col-md-2 col-sd-2">
        <div class="navbuts">
        <a class="navname" href="/predict">Predict</a>
        </div>
      </div>
      <div class="navbut col-lg-1 col-md-2 col-sd-2">
        <div class="navbuts">
        <a class="navname" href="/login">Login</a>
        </div>
      </div>
    </div>
    <div class="mainbody">
      <link rel="stylesheet" type="text/css" href="static/register.css">
      <div class=body>
        <h1 class="sign">Sign  Up</h1>
        <form action="{{ url_for('register') }}" method="post">
            <div class="msg">{{ msg }}</div>
                <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/></br></br>
              <input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox"/></br></br>
                <input id="email" name="email" type="text" placeholder="Enter Your Email ID" class="textbox"/></br></br>
              <input type="submit" class="btn" value="Sign Up"></br>
          </form>
      </div>
    </div>
```

```
    <div id="footer">
        <a target="_blank" class="linkedinname" href="https://www.linkedin.com/in/rohit-d-898384211/">
        Designed by Rohit D <img class="linkedin" src="/static/linkedin.png">
        </a>
    </div>
</body>
</html>
```

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-19465-1659698319

Project video demonstration