Real-Time Communication System Powered by AI for Specially Abled

TEAM ID: PNT2022TMID01172

- TEAM LEADER : Stanley Deivanayagam N
- TEAM MEMBER1: Sajith M
- TEAM MEMBER2: Sachin Prakash Raj R
- TEAM MEMBER3: Harish Kumar

CONTENTS

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. Real-time communications (RTC) is any mode of telecommunications in which all users can exchange information instantly.

Communication plays a significant role in making the world better place. It creates a bonding and relations among the people.

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 using the convolutional neural network.

An app is built which enables the deaf and dumb people to convey their information using signs which is converted to human understandable language and output is given as speech.

2. LITERATURE SURVEY

2.1 Existing problem

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.

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.

2.2 References

1. Upendran, S., and Thamizharasi, A., "American Sign Language interpreter system for deaf and dumb individuals", In the Proceedings of the International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 1477-1481, 2014

2. Lotti, F.,Tiezzi, P., Vassura, G.,Biagiotti, L., and Melchiorri, C., "UBH 3: an anthropomorphic hand with simplified endo-skeletal structure and soft continuous fingerpads", In Proceedings IEEE International Conference on Robotics and Automation, 2004 (ICRA'04), Vol.5, pp. 4736-474, IEEE, 2004.

3. Rajamohan, A., Hemavathy, R., andDhanalakshmi, M., "Deaf-Mute Communication Interpreter", International Journal of Scientific Engineering and Technology, Vol.2, No.5, pp.336-341, 2013.

4. Verma, P., Shimi S. L. and Priyadarshani, R., "Design of Communication Interpreter for Deaf and Dumb Person", Vol.4, no.1, 2013.

2.3 Problem Statement Definition

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.

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.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

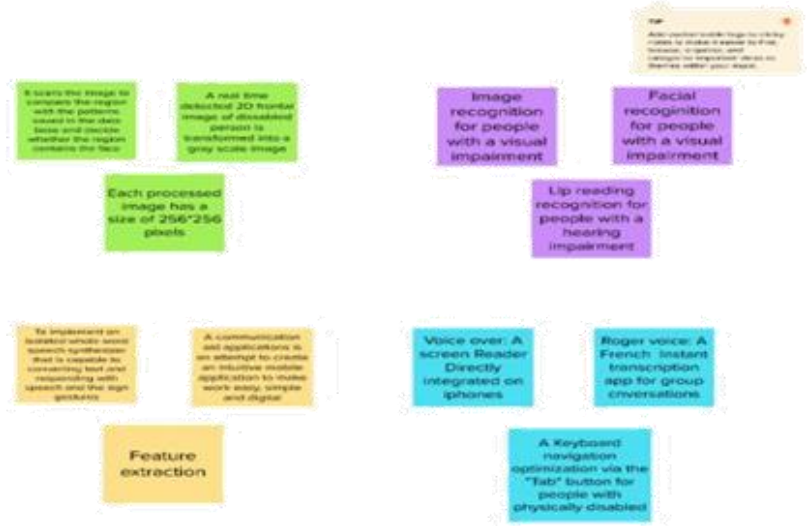
3.2 Ideation & Brainstorming

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

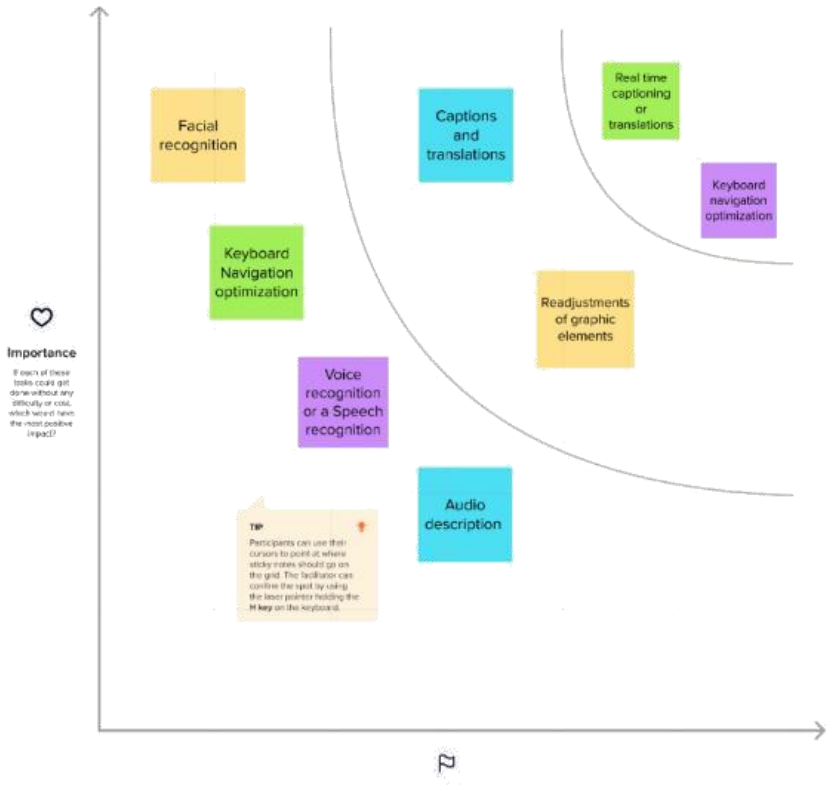


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

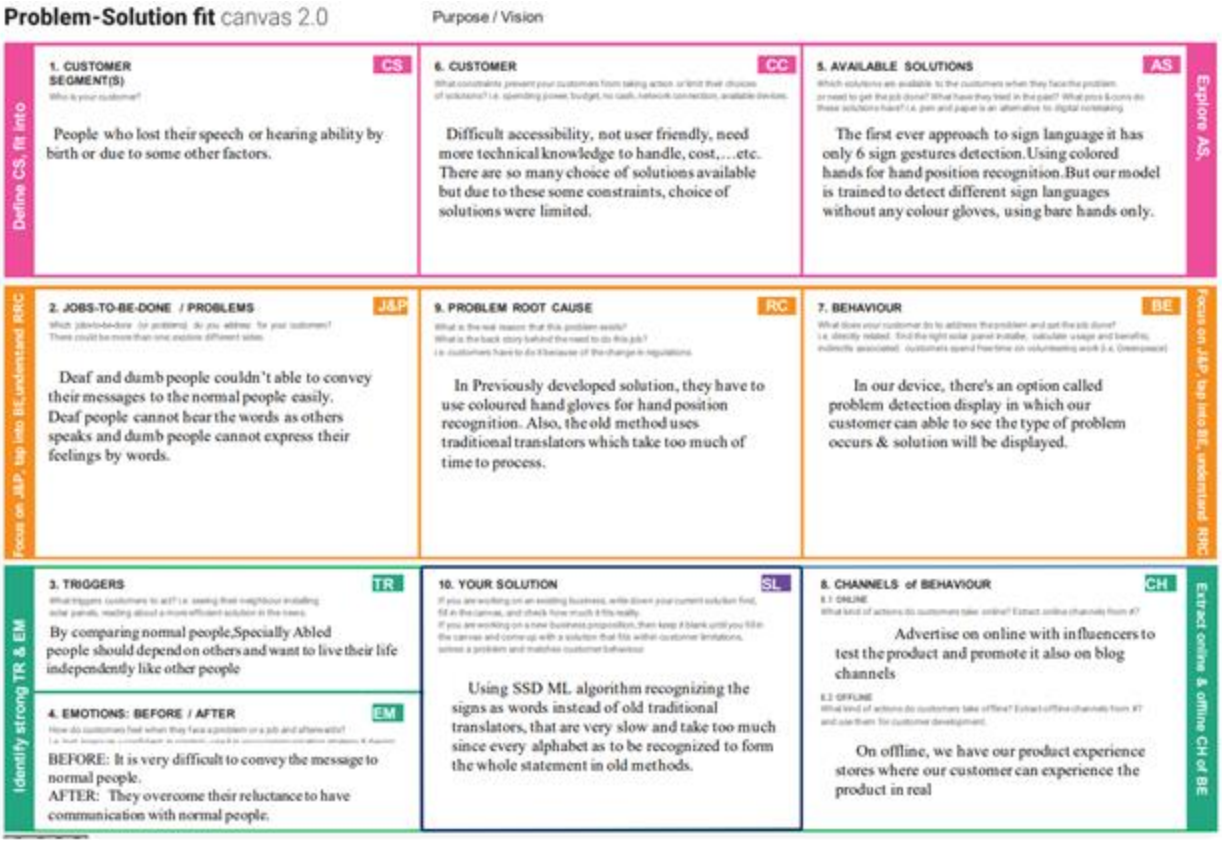


3.3 Proposed Solution

Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Deaf and dumb people couldn't able to communicate with the normal people easily.
2.	Idea/Solution description	A real time ML based system is built for the real time sign language detection with a Tensor Flow object detection.
3.	Novelty/Uniqueness	This model using SSD ML algorithm recognizing the signs as words instead of old traditional translators, that are very slow and take too much since every alphabet as to be recognized to form the whole statement in old methods.
4.	Social Impact/Customer satisfaction	It drastically reduce communication difference gap between normal people and specially abled people with the help of AI. So they can live their life independently.
5.	Business Model (RevenueModel)	We use freemium business revenue model for making revenue. In our device, we give most of the basic features for free of charge but they have to pay if they need more advanced features.
6.	Scalability of the Solution	The model which is TensorFlow model that has been used can be replaced with another model as well. The same system can be implemented for different sign languages by substituting the dataset.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	User Communication	Communication can be done through pc or mobile camera.
FR-4	User requirement	Option should be shown for hand sign to text and voice conversion and vice versa.
FR-5	Communication requirement	Tutor can be made available to have one to one teaching for user.
FR-6	Regulatory requirements	App shutdown in case of cyber attack
FR-7	Reporting	If any issues found in the application, automatically it will be notified to the developer.
FR-8	Compliance to rules or law	Terms and conditions, private policy, End user subscription agreement.

4.2 Non-Functional requirements

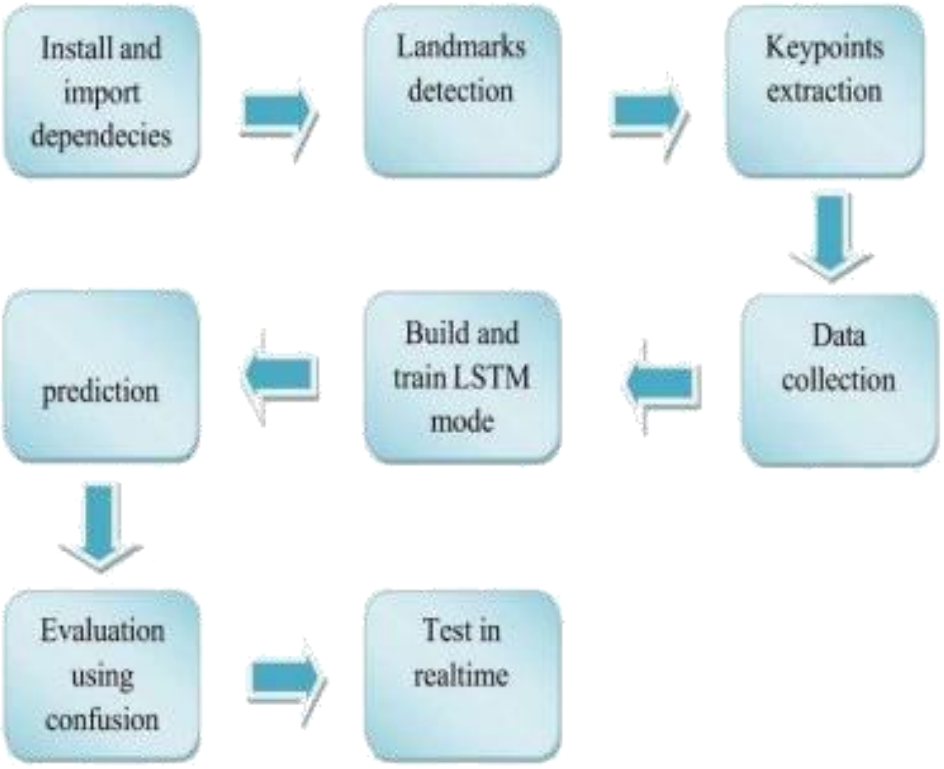
Non-functional Requirements:

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

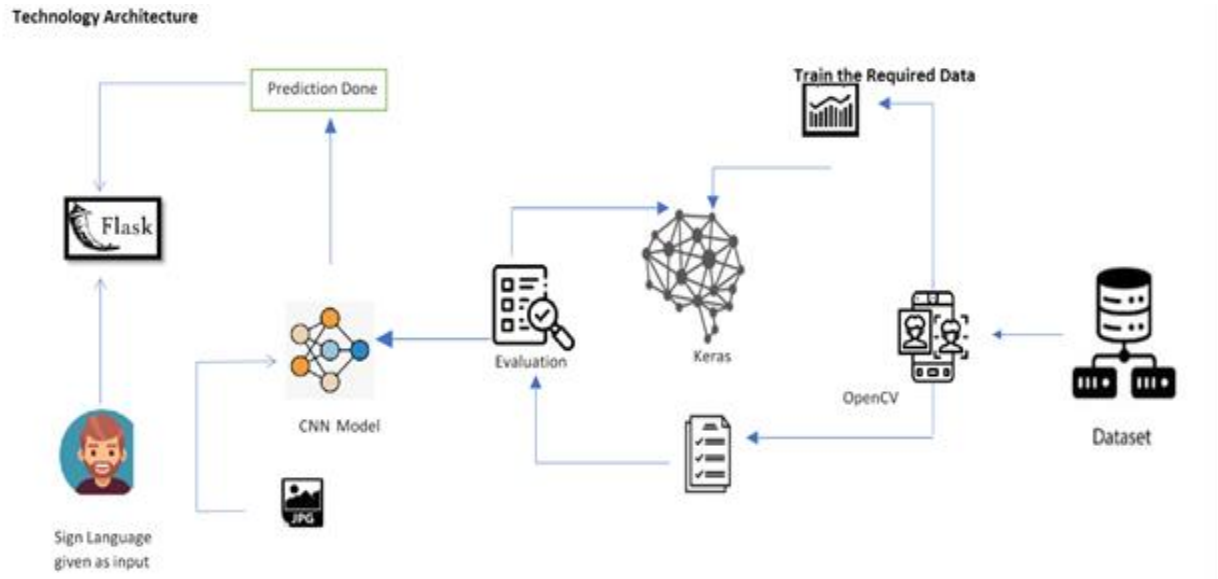
FR No.	Non-Functional Requirements	Description
NFR-1	**Usability**	The camera captures all expressions including facial expressions and hand gestures which can be easily used by all age groups. It can be used by deaf-mute people and their care takers.
NFR-2	**Security**	The system is more secure and information of the customers is also maintained confidentially.
NFR-3	**Reliability**	The system is very liable, it can last for long amounts of time if well maintained.
NFR-4	**Performance**	The performance of the model is efficient. The cost-effective nature of the system makes it extremely liable. The latency is very less for the conversion process.
NFR-5	**Availability**	The solution is suitable for different languages and can be used in many countries. It can be trained for all the available sign languages. This model can be used at any time anywhere.
NFR-6	**Scalability**	The system gives output rapidly. It also predicts quickly when it gets so many inputs at a time. It predicts different types of sign language at a time. Upto 25000 users can be use this model at a time.

5. PROJECT DESIGN

5.1 Data Flow Diagram



5.2 Solution & Technical Architecture



5.3 User Stories

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Data Collection	USN-1	Collect the dataset from alphabet A to Z.	2	Medium
Sprint-1	Training Dataset	USN-2	Train the collected dataset to identify the alphabet	3	High
Sprint-2	Testing the trained model	USN-3	To check whether the data got trained we do the testing for the trained model	1	low
Sprint-2	Saving the text	USN-4	Capture each alphabet and form it as a text and saving it	2	Medium
Sprint-3	Building application	USN-5	Build the flask application and Html page	3	High
Sprint-4	Integrate flask with test code	USN-6	Integrate the flask application with the test code	2	Medium
Sprint-4	Convert text to speech	USN-7	After capturing the text in the application convert to speech	3	High

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Milestone Activity Plan.

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 libraries and 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, Adding the 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 resulting the final output.

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect the dataset from alphabet A to Z.	2	Medium	Harish Kumar
Sprint-1	Training Dataset	USN-2	Train the collected dataset to identify the alphabet	3	High	Stanley Deivanayagam N
Sprint-2	Testing the trained model	USN-3	To check whether the data got trained we do the testing for the trained model	1	low	Sachin Prakash Raj. R
Sprint-2	Saving the text	USN-4	Capture each alphabet and form it as a text and saving it	2	Medium	Sajith. M
Sprint-3	Building application	USN-5	Build the flask application and Html page	3	High	Sajith. M
Sprint-4	Integrate flask with test code	USN-6	Integrate the flask application with the test code	2	Medium	.Sachin Prakash Raj. R, Harish Kumar
Sprint-4	Convert text to speech	USN-7	After capturing the text in the application convert to speech	3	High	Sajith. M, Stanley Deivanayagam N

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

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

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[ ] ls

    drive/  sample_data/

[ ] cd/content/drive/MyDrive/nalaiyathiran

    /content/drive/MyDrive/nalaiyathiran

[ ] ls

    A2Zdata/  A2Zdata.h5  A2Zdata.zip

[ ] pwd

    '/content/drive/MyDrive/nalaiyathiran'

[ ] !unzip A2Zdata.zip

    Streaming output truncated to the last 5000 lines.
      inflating: A2Zdata/training/H/Image_1666640068.7893028.jpg
      inflating: A2Zdata/training/H/Image_1666640069.0229783.jpg
      inflating: A2Zdata/training/H/Image_1666640069.2740915.jpg
      inflating: A2Zdata/training/H/Image_1666640069.4986484.jpg
      inflating: A2Zdata/training/H/Image_1666640069.732095.jpg
```

```
[ ]  inflating: A2Zdata/training/I/Image_1666640123.3193011.jpg
     inflating: A2Zdata/training/I/Image_1666640123.6062934.jpg
     inflating: A2Zdata/training/I/Image_1666640123.9316785.jpg
     inflating: A2Zdata/training/I/Image_1666640124.285531.jpg
     inflating: A2Zdata/training/I/Image_1666640124.621861.jpg
     inflating: A2Zdata/training/I/Image_1666640124.9734807.jpg
     inflating: A2Zdata/training/I/Image_1666640125.2637064.jpg
     inflating: A2Zdata/training/I/Image_1666640127.0510988.jpg
     inflating: A2Zdata/training/I/Image_1666640127.4718907.jpg
     inflating: A2Zdata/training/I/Image_1666640127.7670643.jpg
     inflating: A2Zdata/training/I/Image_1666640128.3012989.jpg
     inflating: A2Zdata/training/I/Image_1666640129.3945134.jpg
     inflating: A2Zdata/training/I/Image_1666640129.7399273.jpg
     inflating: A2Zdata/training/I/Image_1666640131.0610347.jpg
     inflating: A2Zdata/training/I/Image_1666640131.5184145.jpg
```

Image Augmentation

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

```
[ ] train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True, vertical_flip=False)
```

```
[ ] test_datagen= ImageDataGenerator(rescale=1./255)
```

```
[ ] x_train = train_datagen.flow_from_directory(r"/content/drive/MyDrive/nalaiyathiran/A2Zdata/training",target_size=(100,100),class_mode='categorical',batch_size=75)
```

```
Found 7132 images belonging to 26 classes.
```

```
[ ] x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/nalaiyathiran/A2Zdata/testing",target_size=(100,100),class_mode='categorical',batch_size=75)
```

```
Found 2862 images belonging to 26 classes.
```

```
[ ] x_train.class_indices
```

```
{'A': 0,
 'B': 1,
 'C': 2,
 'D': 3,
 'E': 4,
 'F': 5,
 'G': 6,
 'H': 7,
 'I': 8,
 'J': 9,
 'K': 10,
 'L': 11,
 'M': 12,
 'N': 13,
 'O': 14,
 'P': 15,
 'Q': 16,
 'R': 17,
 'S': 18,
 'T': 19,
 'U': 20,
 'V': 21,
 'W': 22,
 'X': 23,
 'Y': 24,
 'Z': 25}
```

Model

```
[ ] from tensorflow.keras.models import Sequential
```

Layers

Layers

```
[ ] from tensorflow.keras.layers import Dense, Convolution2D, MaxPooling2D, Flatten
```

```
[ ] model = Sequential()
```

```
[ ] model.add(Convolution2D(32, (3,3), input_shape=(100,100,3),activation = 'relu')) #Feature map
```

```
[ ] model.add(MaxPooling2D(pool_size = (2,2))) #Pooled matrix
```

```
[ ] model.add(Flatten())
```

```
[ ] model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 98, 98, 32)        896

 max_pooling2d (MaxPooling2D  (None, 49, 49, 32)       0
 )

 flatten (Flatten)           (None, 76832)             0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

```
[ ] model.add(Dense(512,activation='relu'))
    model.add(Dense(256,activation='relu'))
```

```
[ ] model.add(Dense(26,activation='softmax'))
```

Compile

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

```
[ ] len(x_train)
```

```
96
```

```
len(x_test)
```

```
39
```

Fit the Model

```
[ ] model.fit_generator(x_train, steps_per_epoch=len(x_train), validation_data=x_test, validation_steps=len(x_test),epochs=5)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which
  """Entry point for launching an IPython kernel.
Epoch 1/5
96/96 [==============================] - 168s 2s/step - loss: 2.4762 - accuracy: 0.5006 - val_loss: 0.7348 - val_accuracy: 0.7628
Epoch 2/5
96/96 [==============================] - 147s 2s/step - loss: 0.3840 - accuracy: 0.8850 - val_loss: 0.4430 - val_accuracy: 0.9113
Epoch 3/5
96/96 [==============================] - 147s 2s/step - loss: 0.1977 - accuracy: 0.9397 - val_loss: 0.4319 - val_accuracy: 0.9238
Epoch 4/5
```

```
Epoch 4/5
96/96 [==============================] - 145s 2s/step - loss: 0.1413 - accuracy: 0.9586 - val_loss: 0.3511 - val_accuracy: 0.9382
Epoch 5/5
96/96 [==============================] - 149s 2s/step - loss: 0.0954 - accuracy: 0.9700 - val_loss: 0.2950 - val_accuracy: 0.9486
<keras.callbacks.History at 0x7fce93420890>
```

Save the model

```
[ ] model.save('A2Z.h5')
```

```
[ ] ls
```

```
A2Zdata/  A2Zdata.h5  A2Zdata.zip  A2Z.h5
```

Test the model

```
[ ] import numpy as np
    from tensorflow.keras.models import load_model
```

```
[ ] from tensorflow.keras.preprocessing import image
```

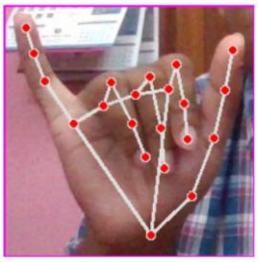
```
[ ] model=load_model('A2Z.h5')
```

```
[ ] pwd
```

```
'/content/drive/MyDrive/nalaiyathiran'
```

```
[ ] img=image.load_img(r'/content/drive/MyDrive/nalaiyathiran/A2Zdata/testing/Y/Image_1667328891.1069646.jpg')
```

```
[ ] img
```



```
[ ] img=image.load_img(r'/content/drive/MyDrive/nalaiyathiran/A2Zdata/testing/Y/Image_1667328891.1069646.jpg',target_size=(100,100))
```

```
[ ] img
```

```
[ ] x=image.img_to_array(img)
```

```
[ ] x
```

```
[ ] x
```

```
array([[[196.,  40., 201.],
        [247.,   7., 238.],
        [240.,  14., 235.],
        ...,
        [238.,  13., 231.],
        [234.,  16., 224.],
        [249.,   6., 248.]],

       [[208.,  34., 207.],
        [243., 132., 200.],
        [219., 146., 175.],
        ...,
        [144.,  99., 102.],
        [143., 110.,  91.],
        [204.,  56., 178.]],

       [[202.,  36., 206.],
        [241., 144., 195.],
        [224., 150., 185.],
        ...,
        [149., 110.,  81.],
        [148., 110.,  87.],
        [205.,  58., 173.]],

       ...,

       [[203.,  37., 199.],
        [207., 118., 176.],
        [192., 131., 149.],
        ...,
        [116., 133., 141.],
        [138., 166., 170.],
        [209.,  93., 226.]],

       [[206.,  32., 207.],
        [216., 118., 179.],
        [186., 128., 140.],
        ...,
        [220.,  82., 229.],
        [237., 118., 246.],
        [255.,  62., 255.]]], dtype=float32)
```

```
[ ] x.shape
```

```
(100, 100, 3)
```

```
[ ] x= np.expand_dims(x,axis=0)
```

```
[ ] x
```

```
array([[[[196.,  40., 201.],
         [247.,   7., 238.],
         [240.,  14., 235.],
         ...,
         [238.,  13., 231.],
         [234.,  16., 224.],
         [249.,   6., 248.]],

        [[208.,  34., 207.],
         [243., 132., 200.],
         [219., 146., 175.],
         ...,
         [144.,  99., 102.],
         [143., 110.,  91.],
         [204.,  56., 178.]],

        [[202.,  36., 206.],
         [241., 144., 195.],
         [224., 150., 185.],
         ...,
         [149., 110.,  81.],
         [148., 110.,  87.],
         [205.,  58., 173.]],
```

```
[ ]          [238.,  62., 222.],
             [239.,  59., 220.],
             ...,
             [220.,  82., 229.],
             [237., 118., 246.],
             [255.,  62., 255.]]], dtype=float32)
```

```
[ ] x.shape

    (1, 100, 100, 3)
```

```
[ ] y= np.argmax(model.predict(x),axis=1)

    1/1 [==============================] - 0s 66ms/step
```

```
[ ] y

    array([24])
```

```
[ ] x_train.class_indices

    {'A': 0,
     'B': 1,
     'C': 2,
     'D': 3,
     'E': 4,
     'F': 5,
     'G': 6,
     'H': 7,
     'I': 8,
     'J': 9,
     'K': 10,
     'L': 11,
     'M': 12,
     'N': 13,
```

```
x_train.class_indices

{'A': 0,
 'B': 1,
 'C': 2,
 'D': 3,
 'E': 4,
 'F': 5,
 'G': 6,
 'H': 7,
 'I': 8,
 'J': 9,
 'K': 10,
 'L': 11,
 'M': 12,
 'N': 13,
 'O': 14,
 'P': 15,
 'Q': 16,
 'R': 17,
 'S': 18,
 'T': 19,
 'U': 20,
 'V': 21,
 'W': 22,
 'X': 23,
 'Y': 24,
 'Z': 25}
```

```
[ ] index=['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
```

```
[ ] index[y[0]]

    'Y'
```

8. TESTING

8.1 Test Cases

```
[ ] img=image.load_img(r'/content/drive/MyDrive/nalaiyathiran/A2Zdata/training/B/Image_1666335447.728338.jpg',target_size=(100,100))
```

```
img
```

```
[ ] x=image.img_to_array(img)
```

```
[ ] x=np.expand_dims(x,axis=0)
```

```
[ ] y=np.argmax(model.predict(x),axis=1)

    1/1 [==============================] - 0s 56ms/step
```

```
[ ] y

    array([1])
```

```
[ ] index=['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
```

```
[ ] index[y[0]]

    'B'
```

8.2 User Acceptance Testing

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	11	2	3	2	18
Duplicate	1	3	4	0	8
External	3	5	0	0	8
Fixed	12	2	5	22	41
Not Reproduced	0	1	0	0	1
Skipped	0	0	1	2	3
Won't Fix	0	4	1	1	7
Totals	27	17	14	27	86

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	49	0	0	49
Security	4	0	0	4
Outsource Shipping	4	0	0	4
Exception Reporting	11	0	0	11
Final Report Output	2	0	0	2
Version Control	1	0	0	1

9. RESULTS

9.1 Performance Metrics

Technical Skills Evaluation Metrics								
S. No.	Metrics	Weightage (%)	Description	Sub-Evaluation Metrics & Scoring Criteria	Score	Description	Standard Template	Scoring Guidelines
1	**Technical Training & Assignments**	5%	This metric will be scored based on the following 1. The Evaluation is the Quiz at the end of each training session 2. Submission of completed assignments	**Number of Tests / Quizzes Attempted (Total = 4 Nos.)** **Assignment-1** **Assignment-2** **Assignment-3** **Assignment-4**	Quiz will be of MCQ type with 10 questions 1 2 3 4	Weekly Assignment Weekly Assignment Weekly Assignment Weekly Assignment	No	Each Quiz Carries – 2.5 Score Completed – 2, Partial = 1, Not Submitted -0 Completed – 2, Partial = 1, Not Submitted -0 Completed – 2, Partial = 1, Not Submitted -0 Completed – 2, Partial = 1, Not Submitted -0
2	**Innovation & Problem Solving**	15%	This metric will be assessed and scored, based on the unique innovative features/ work-out in the proposed solution. The team shall submit the following templates: 1. Define the Problem Statement 2. Empathize & Discover 3. Brainstorm & Prioritize Ideas	**1. Empathize & Discover**	5	A problem statement often touches on the 5 w's (who, what, where, when, why) of the problem. Submit the template provided	Yes	Completed – 2, Partial = 1, Not Submitted -0
				2. Define the Problem Statement	4	When designing a solution to a problem, it's important to understand who you're design it for. Are you can keep their needs, challenges, behaviors, and attitudes in mind while you design	Yes	Completed – 4, Partial -2, Not Submitted -0
				3. Brainstorm & Prioritize Ideas	4	Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving.	Yes	Completed – 4, Partial -2, Not Submitted -0
3	**Requirement Analysis using Critical Thinking**	10%	This metric will be scored based on the Submission of Project Requirement Analysis Template with valid information	**Determine the Requirements (Customer Journey Maps)**	4	Create customer journey maps to understand the requirements of customer	Yes	Completed – 4, Partial -2, Not Submitted -0
				Requirement Analysis (Functional, Operational, Technical) / Flow Charts	4	Analyze the requirements, categorize them into Functional, Operational and technical. Flow charts can be prepared for each requirement.	Yes	Completed – 4, Partial -2, Not Submitted -0
				Interpretation & Modeling	2	Decide which of the categorized requirements can be achieved and how they will be delivered. Find out and document what is expected or wanted from the product, project, or application	No	Completed – 2, Partial -1, Not Submitted -0
4	**Process Design using Design Thinking**	10%	This metric will be assessed and scored, based on the submission of project design templates which includes: **1. Proposed solution** - Novelty - Feasibility of Concept - Social / Business Impact **2. Problem - Solution Fit** **3. Solution Architecture**	**Proposed solution** - Novelty - Feasibility of Concept - Social / Business Impact	5	Shall include the uniqueness of proposed solution, novelty, social or business impact and feasibility of solution	Yes	Completed – 5, Partial -1, Not Submitted -0
				Problem - Solution Fit	4	It is a template to help identify solutions with high relevance of solution adoption, not-a-or time-spent on solving, and get a better overview of the current situation	Yes	Completed – 3, Partial -1, Not Submitted -0
				Solution Architecture	4	Shall cover the architectural aspect of project from concept to deployment	Yes	Completed – 2, Partial -1, Not Submitted -0
5	**Technology Stack**	5%	This metric will be assessed and scored, based on the selection of technology in building the proposed solution.	**Technical Architecture:** **- Is the system robust** **- Is it highly modifiable?** **- Is it scalable?** **- Is it testable (wrt time & budget constraints)**	4	Detailed technical architecture of the solution shall be developed. Samples of code patterns will be introduced here for the reference	No	Not Submitted – 0, Partial – 1 to 3 , Complete - 4
				Open Source Frameworks	3	Assess the utilization of open source frameworks in the solution	No	Open source framework used: Only one - 1 Mark More than 3 - 3 marks
				Third party API's	3	Assess the number of API's integrated in the solution	No	1 -2 API's - 1 Mark 3 API's - 3 Marks
				Cloud Deployment	3	Check whether the solution deployed on cloud or developed locally. Also check have they implemented auto scaling and rollback etc.	No	Simple Architecture – 1 Automated Containers – 3
6	**Project Planning using Agile Methodologies**	5%	This metric will be assessed and scored, based on the detailed project planning template submitted by the project team. The template shall cover 1. Project Milestones & Tasks 2. Sprint Delivery Schedule 3. Distribution of Tasks among team members 4. Team management tool for agile planning	**Project Milestones & Tasks**	4	Ensure the detailed planning of project development, list down important features, required roles, amount of work, list stories & epics, list iterations	Yes	Not Submitted – 0, Partial – 1 to 3 , Complete - 4
				Sprint Delivery Plan	3	Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint	Yes	Number of Sprint planning meetings organized, Minutes of meeting recorded Sprint not Held 0 – 0, 0.5 – 10 , 1 –>10 , 2
				Project Progress Tracking	4	Distribution of tasks to team members, Prioritization of tasks, Track the progress of all tasks, Submit velocity & BURN DOWN CHART	No	Not Submitted – 0, Partial – 1, Complete- 2
				Team management tool for agile planning (e.g. Trello, Jira, etc.)	3	Utilization of project / team management tools for the project delivery	No	No tools used – 0 marks Any one tool – 3 marks
7	**Coding & Scheduling**	15%	This metric will be scored based on the following submission of the code: 1.No. of Functional Features 2.Code Efficiency 3.Utilization of algorithms 4.Coding Layout / Self-explanation & re-usability 5.Debugging & Traceability 6.Dynamic Flexibility	**No. of Functional Features included in the solution**	2	Assess the feature presented vs. requirement	No	0 – 50% Mapping - 0 Marks 51 – 85 % Mapping - 1 Mark >=85% Mapping - 2 Marks
				Code Layout, Readability and Reusability	3	Assess the code layout, readability and reusability	No	Average - 0, Good - 1, Excellent - 2
				Utilization of Algorithms, Dynamic Programming, Optimal Memory Utilization	3	Assess the Utilization of Algorithms, Dynamic Programming, Optimal Memory Utilization	No	Average - 0, Good - 1, Excellent - 2
				Debugging & Traceability	3	Assess the Debugging & Traceability Capability in the code	No	Average - 0, Good - 1, Excellent - 2
				Exception Handling	3	Assess the Exception Handling capability of the code developed	No	Average - 0, Good - 1, Excellent - 2
8	**Acceptance Testing**	5%	This metric will be assessed and scored, based on the following: 1.UAT Initiation 2.UAT Design 3.UAT Execution 4.UAT Report Submission 5.Utilization of Test Management Tools	**UAT Initiation**	2	Ensure the availability of UAT checklist (Define Approach, test data requirements, etc.)	No	Not Submitted – 0, Partial – 1, Complete - 2
				UAT Design	2	Assess the UAT design document	No	Not Submitted – 0, Partial – 1, Complete - 2
				UAT Execution	3	Assess the list of bugs identified and fixes implemented	No	Not Submitted – 0, Partial – 1, Complete - 2
				UAT Report Submission	2	Assess the report submitted	Yes	Not Submitted – 0, Partial – 1, Complete - 2
				Utilization of Test management Tools	3	Assess the utilization of Tools like Qtest, Apptest, Load e.t.c etc.	No	Not Submitted – 0, Partial – 1, Complete - 2
9	**Performance Testing**	5%	This metric will be assessed and scored, based on the following: 1.Application Performance (for Cloud & IoT Projects) 2.Model Performance (for ML /AI Projects) 3.Dashboard & User Stories (for Data Analytics Projects)	**Application Performance Metrics** Error rates Response times Account visits Customer experience	10	Assess the application performance metrics submitted or verify using performance testing tools	Yes	Below Average - 0 - 25% Satisfactory – 26% - 50% Good – 51% - 75% Excellent – 76% - 100%
				Model Performance Metrics Confusion matrix Accuracy Precision Recall Specificity F1 score Precision-Recall or PR curve ROC (Receiver Operating Characteristic) curve PR or ROC curve	10	Assess the model performance metrics submitted or verify using model performance testing tools	Yes	Below Average - 0 - 25% Satisfactory – 26% - 50% Good – 51% - 75% Excellent – 76% - 100%
				Dashboard & User Story Metrics Dashboard design Data Representation Relevant Data is Rendered Utilization of Data Filters Calculation Efficiency Effective User Story	10	Assess the visualization dashboard, based on the metrics provided	Yes	Below Average - 0 - 25% Satisfactory – 26% - 50% Good – 51% - 75% Excellent – 76% - 100%

10. ADVANTAGES & DISADVANTAGES

Advantages:

- It is possible to create a mobile application to bridge the communication gap between deaf and dumb persons and the general public.
- As different sign language standards exist, their dataset can be added, and the user can choose which sign language to read.

Disadvantages:

- Also accuracy depends upon distance between camera and object.
- It takes a lot of time to listen, speak, read, or write to someone.

11. CONCLUSION

The proposed communication system between Deaf and Dumb people and ordinary people are aiming for it when bridging the communication gap between two societies. It provides complete two - sided communication in an efficient manner between the disabled and the normal person.

For communication between deaf person and a second person, a mediator is required to translate sign language of deaf person. But a mediator is required to know the sign language used by deaf person. But this is not always possible since there are multiple sign languages for multiple languages.

So to understand all sign languages, Hand gestures of deaf peoples by normal peoples this system is proposed.

12. FUTURE SCOPE

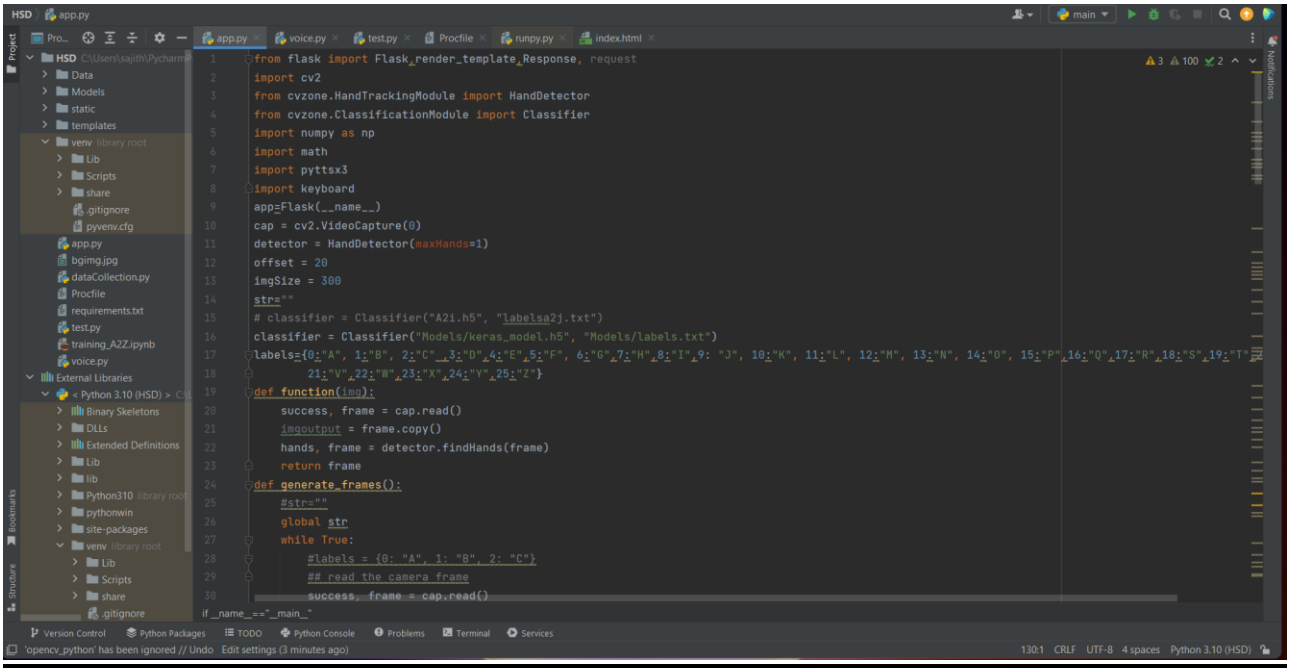
The speech-to-text and text-to-speech technologies helped those people who had difficulties in communicating or expressing their feelings to the normal people.

This reduces the communication gap between the normal people and the specially abled people.

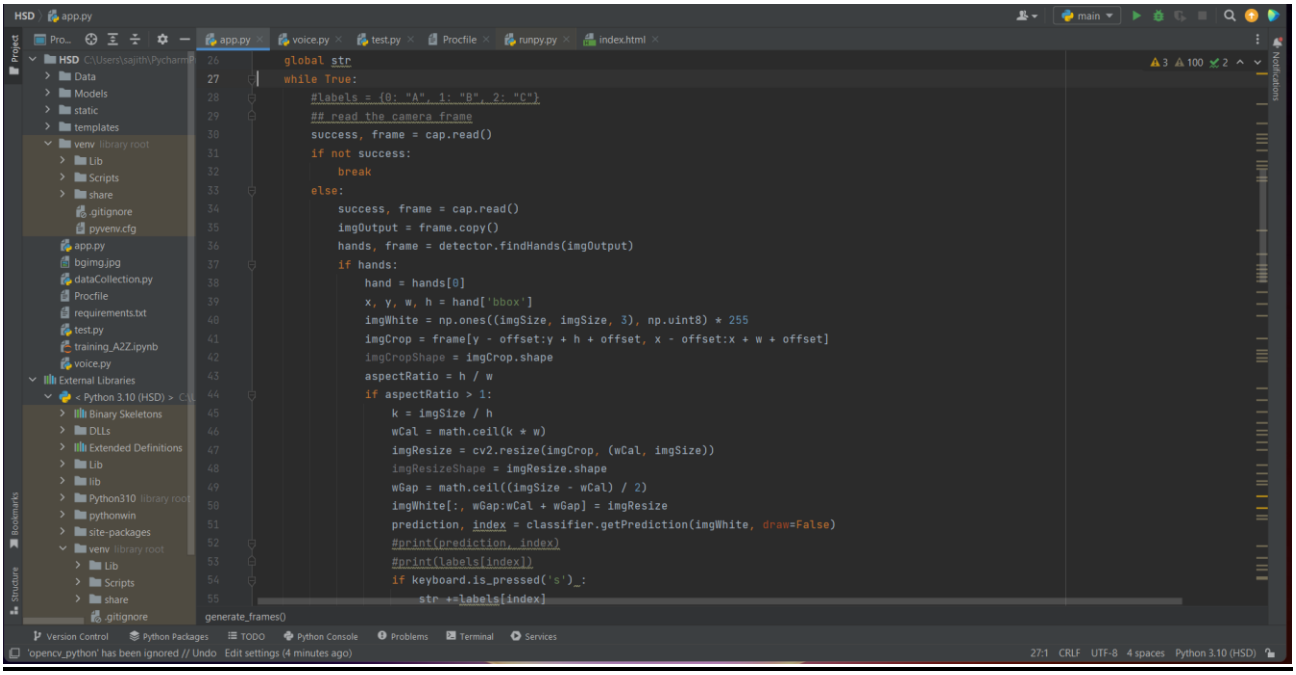
Using image pre-processing and Artificial Intelligence it is easy to understand the context of objects and clearly explains it to the people who use it for communication.

13.APPENDIX

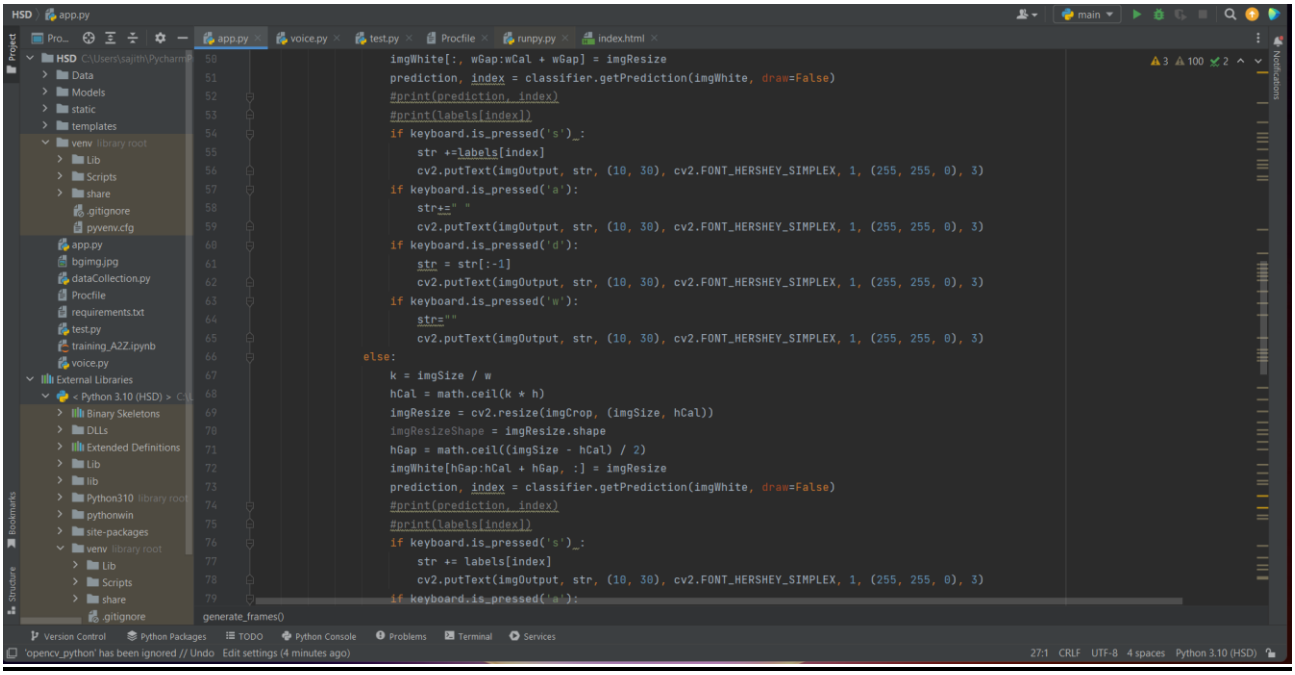
Source Code



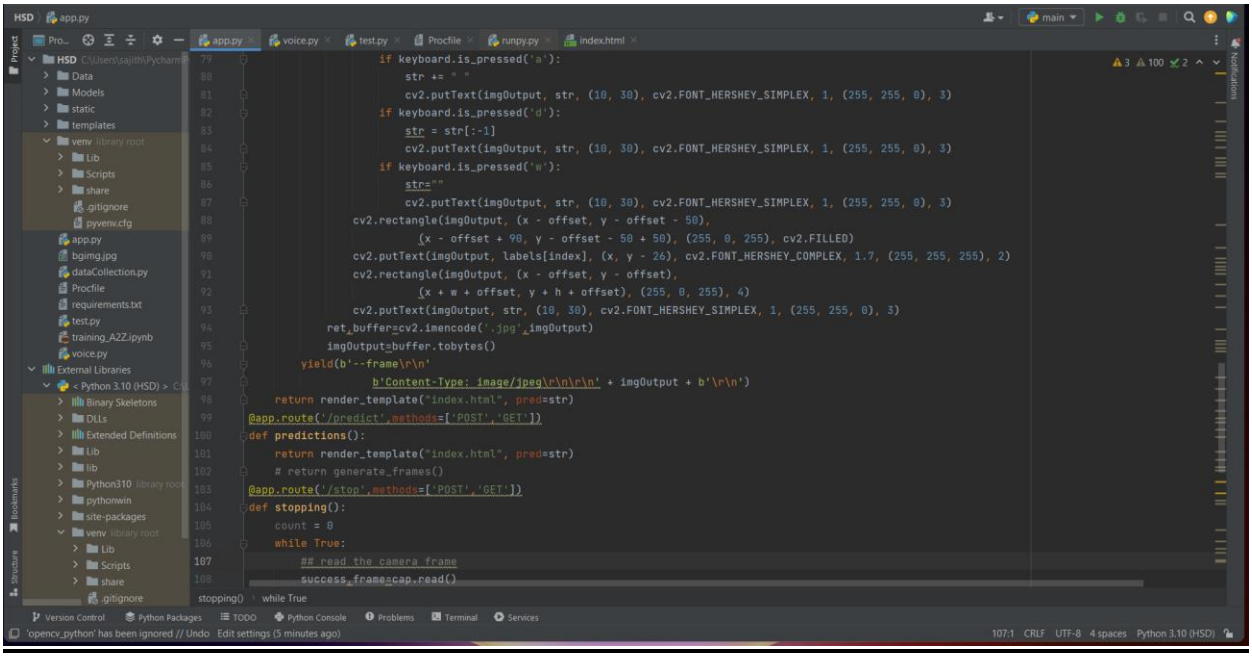
```
from flask import Flask,render_template,Response, request
import cv2
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import numpy as np
import math
import pyttsx3
import keyboard
app=Flask(__name__)
cap = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1)
offset = 20
imgSize = 300
str=""
# classifier = Classifier("A2i.h5", "labelsa2j.txt")
classifier = Classifier("Models/keras_model.h5", "Models/labels.txt")
labels={0:"A", 1:"B", 2:"C",3:"D",4:"E",5:"F", 6:"G",7:"H",8:"I",9: "J", 10:"K", 11:"L", 12:"M", 13:"N", 14:"O", 15:"P",16:"Q",17:"R",18:"S",19:"T",
        21:"V",22:"W",23:"X",24:"Y",25:"Z"}
def function(img):
    success, frame = cap.read()
    imgoutput = frame.copy()
    hands, frame = detector.findHands(frame)
    return frame
def generate_frames():
    #str=""
    global str
    while True:
        #labels = {0: "A", 1: "B", 2: "C"}
        ## read the camera frame
        success, frame = cap.read()
if __name__=="__main__"
```



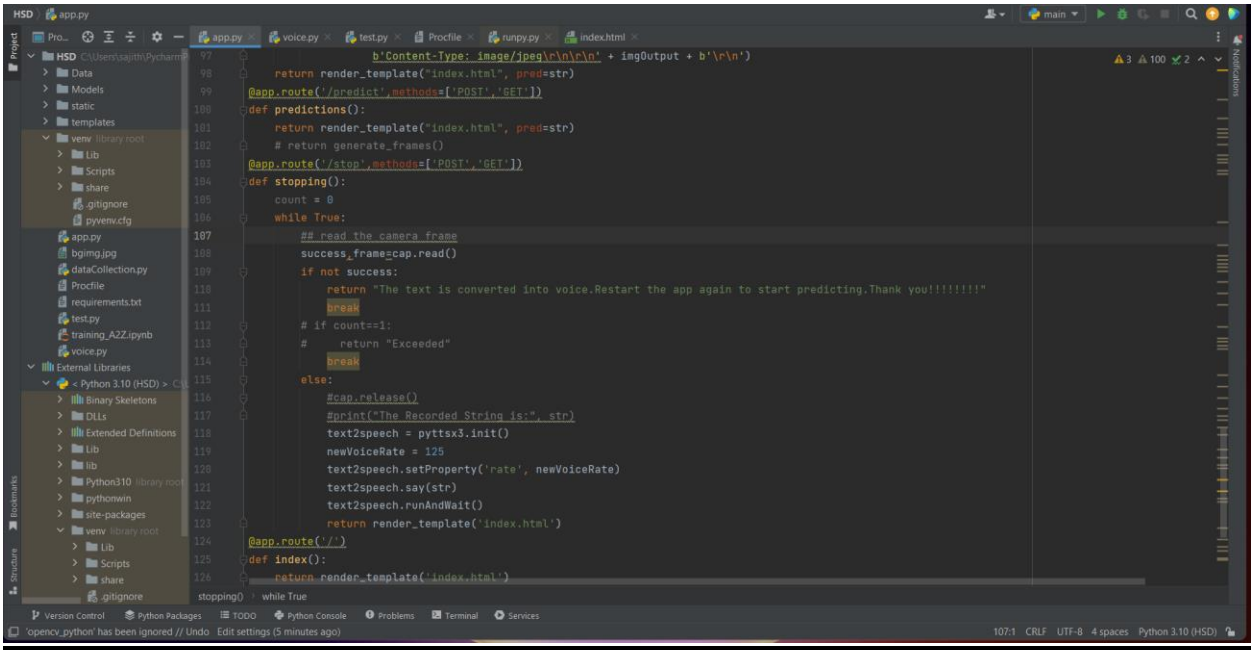
```
    global str
    while True:
        #labels = {0: "A", 1: "B", 2: "C"}
        ## read the camera frame
        success, frame = cap.read()
        if not success:
            break
        else:
            success, frame = cap.read()
            imgOutput = frame.copy()
            hands, frame = detector.findHands(imgOutput)
            if hands:
                hand = hands[0]
                x, y, w, h = hand['bbox']
                imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
                imgCrop = frame[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)
                    #print(labels[index])
                    if keyboard.is_pressed('s'):
                        str +=labels[index]
generate_frames()
```

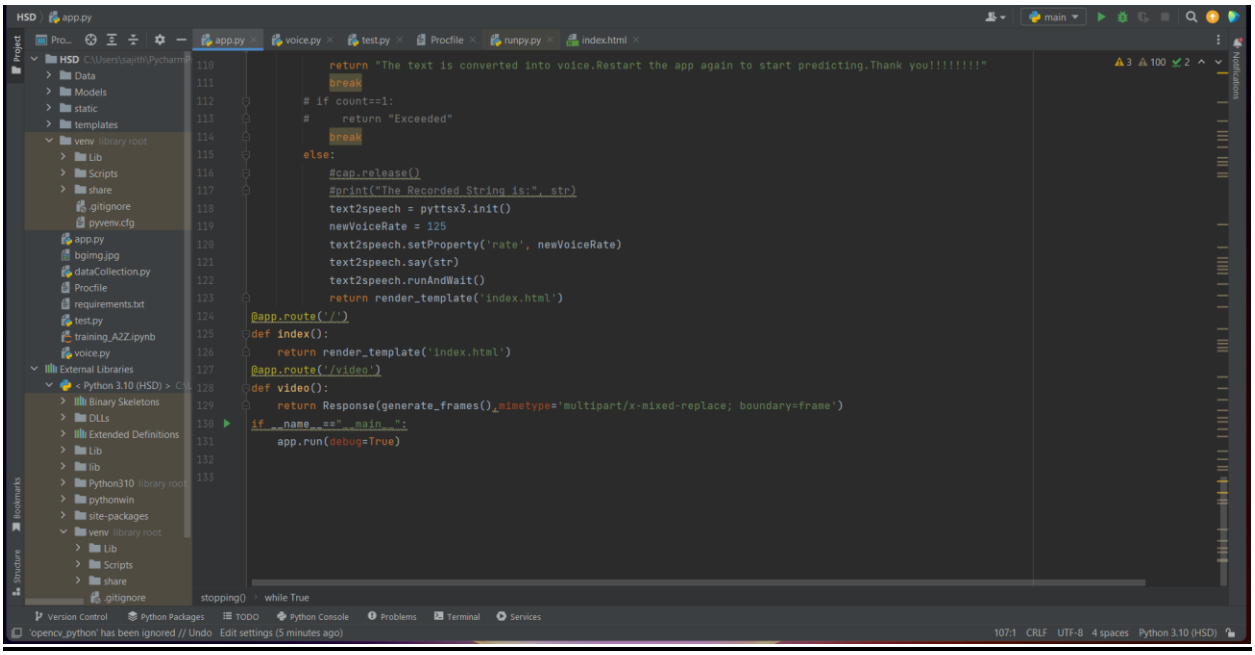


```
                    imgWhite[:, wGap:wCal + wGap] = imgResize
                    prediction, index = classifier.getPrediction(imgWhite, draw=False)
                    #print(prediction, index)
                    #print(labels[index])
                    if keyboard.is_pressed('s'):
                        str +=labels[index]
                        cv2.putText(imgOutput, str, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)
                    if keyboard.is_pressed('a'):
                        str+=" "
                        cv2.putText(imgOutput, str, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)
                    if keyboard.is_pressed('d'):
                        str = str[:-1]
                        cv2.putText(imgOutput, str, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)
                    if keyboard.is_pressed('w'):
                        str=""
                        cv2.putText(imgOutput, str, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)
                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)
                    #print(prediction, index)
                    #print(labels[index])
                    if keyboard.is_pressed('s'):
                        str += labels[index]
                        cv2.putText(imgOutput, str, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)
                    if keyboard.is_pressed('a'):
generate_frames()
```



```
                    if keyboard.is_pressed('a'):
                        str += " "
                        cv2.putText(imgOutput, str, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)
                    if keyboard.is_pressed('d'):
                        str = str[:-1]
                        cv2.putText(imgOutput, str, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)
                    if keyboard.is_pressed('w'):
                        str=""
                        cv2.putText(imgOutput, str, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)
                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.putText(imgOutput, str, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)
            ret,buffer=cv2.imencode('.jpg',imgOutput)
            imgOutput=buffer.tobytes()
        yield(b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + imgOutput + b'\r\n')
    return render_template("index.html", pred=str)
@app.route('/predict',methods=['POST','GET'])
def predictions():
    return render_template("index.html", pred=str)
    # return generate_frames()
@app.route('/stop',methods=['POST','GET'])
def stopping():
    count = 0
    while True:
        ## read the camera frame
        success,frame=cap.read()
stopping() › while True
```





DEMO LINK:

https://youtu.be/OyIJiBZWIKE

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-2475-1658472446