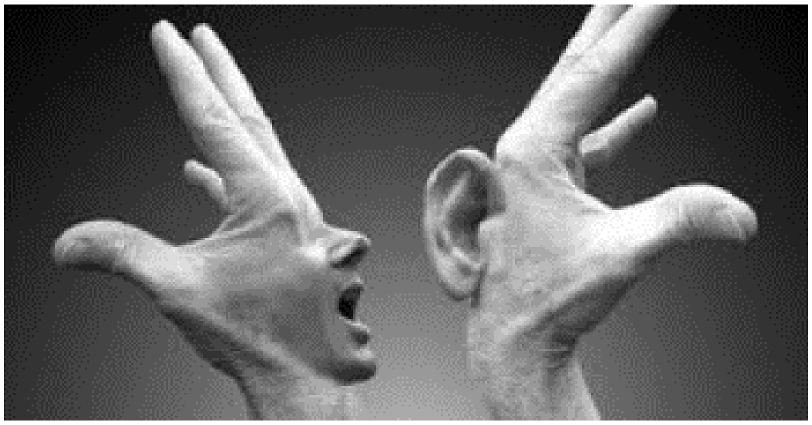
Real-Time Communication System Powered by AI for Specially Abled



TEAM ID: PNT2022TMID00980

1. TEAM LEADER : Dhivagar T
2. TEAM MEMBER1: Balaji G
3. TEAM MEMBER2: Dinesh kumar K
4. TEAM MEMBER3: Dharanidharan D
5. TEAM MEMBER4: Abubakkar Siddiq M

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

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

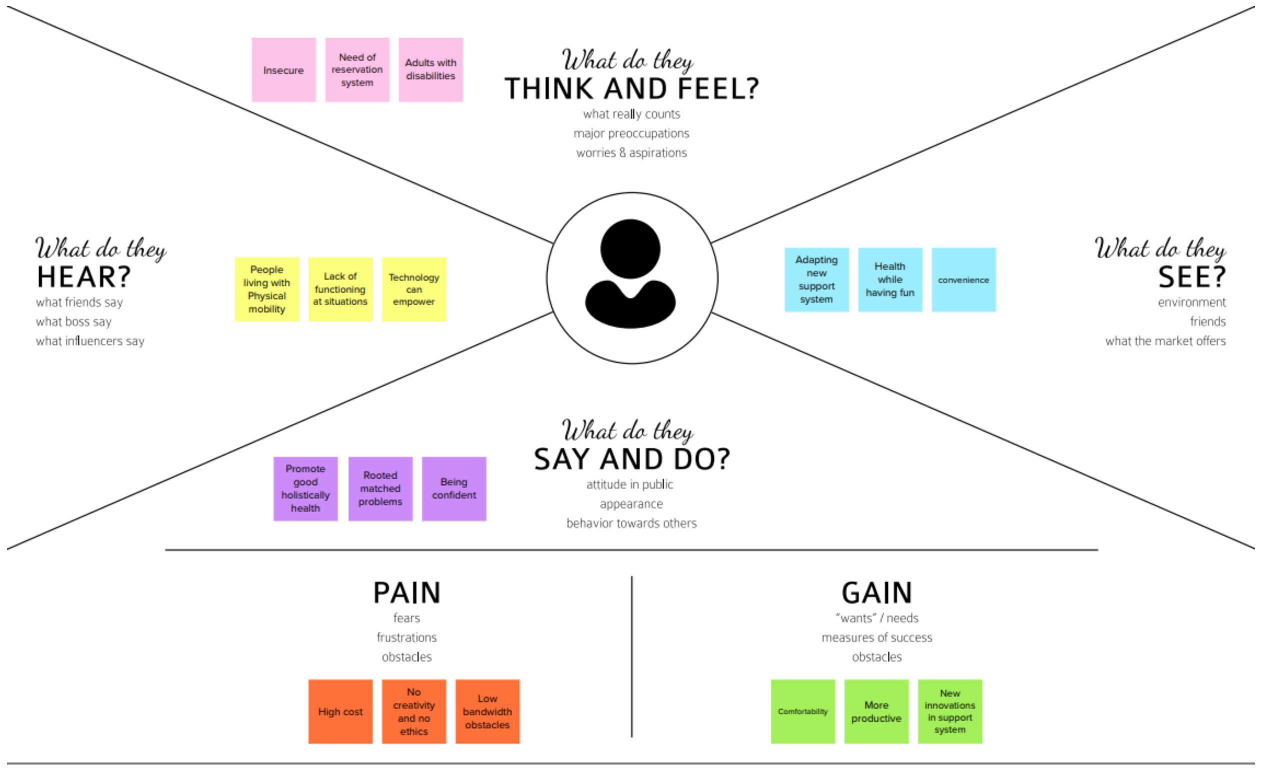
2.3 Problem Statement Definition

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.

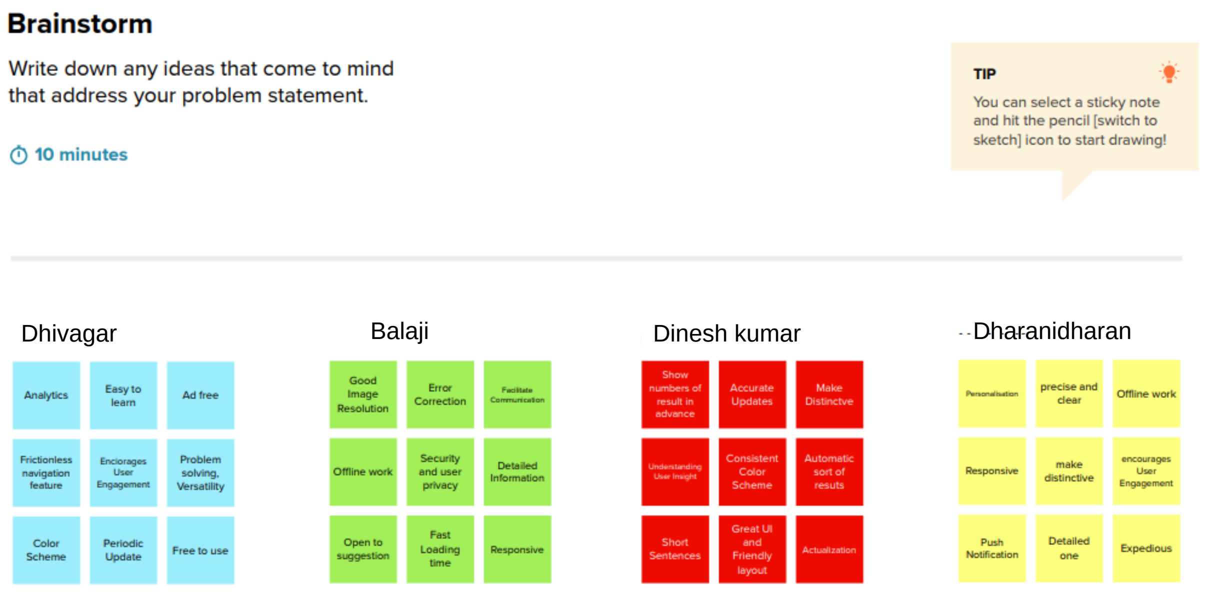
Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 <u>Ideation & Brainstorming</u>



Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

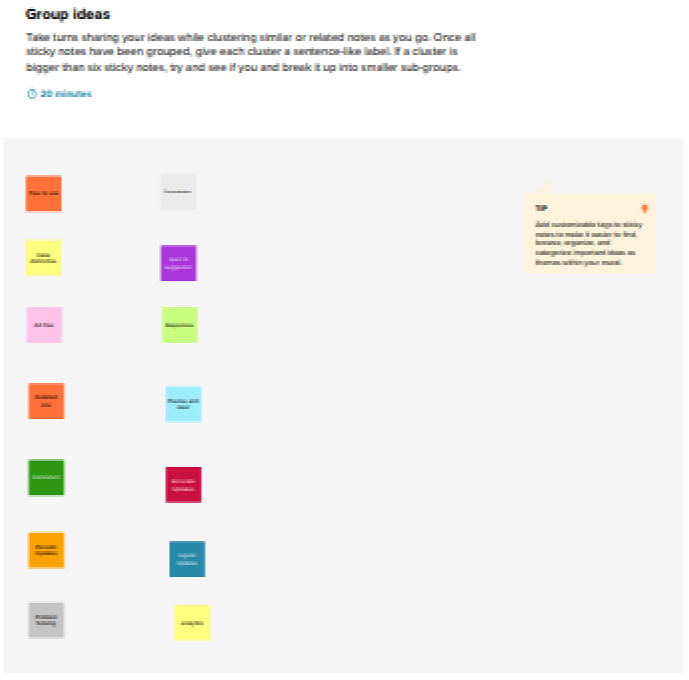
TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Dhivagar		
Analytics	Easy to learn	Ad free
Frictionless navigation feature	Enciorages User Engagement	Problem solving, Versatility
Color Scheme	Periodic Update	Free to use

Balaji		
Good Image Resolution	Error Correction	Facilitate Communication
Offline work	Security and user privacy	Detailed Information
Open to suggestion	Fast Loading time	Responsive

Dinesh kumar		
Show numbers of result in advance	Accurate Updates	Make Distinctve
Understanding User Insight	Consistent Color Scheme	Automatic sort of resuts
Short Sentences	Great UI and Friendly layout	Actualization

Dharanidharan		
Personalisation	precise and clear	Offline work
Responsive	make distinctive	encourages User Engagement
Push Notification	Detailed one	Expedious



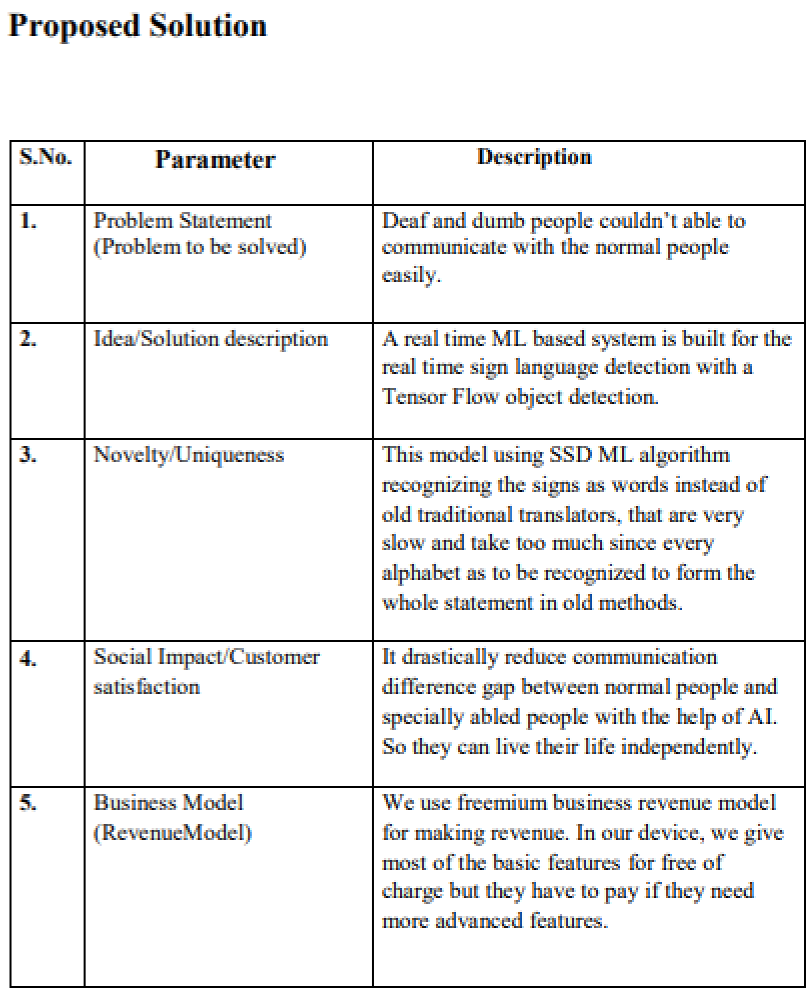
Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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

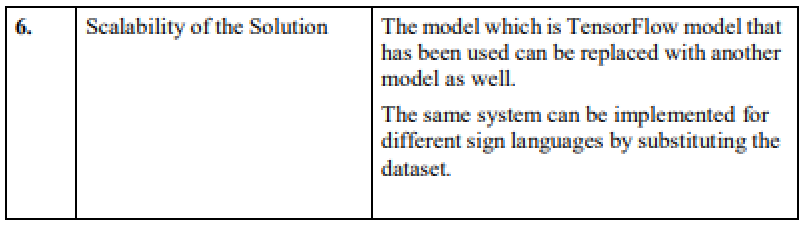
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

3.3Proposed Solution

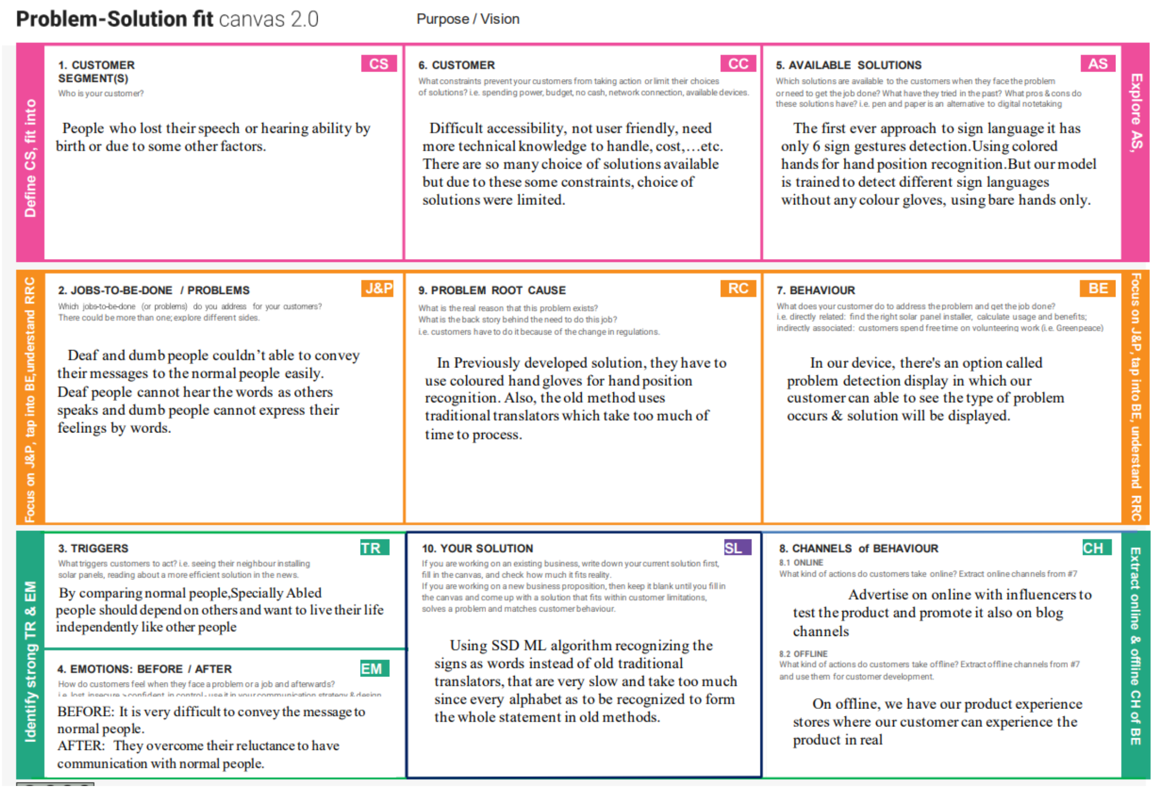


Proposed Solution

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



Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

People who lost their speech or hearing ability by birth or due to some other factors.

6. CUSTOMER CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Difficult accessibility, not user friendly, need more technical knowledge to handle, cost,…etc. There are so many choice of solutions available but due to these some constraints, choice of solutions were limited.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

The first ever approach to sign language it has only 6 sign gestures detection.Using colored hands for hand position recognition.But our model is trained to detect different sign languages without any colour gloves, using bare hands only.

Explore AS,

Focus on J&P, tap into BE,understand RRC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Deaf and dumb people couldn't able to convey their messages to the normal people easily. Deaf people cannot hear the words as others speaks and dumb people cannot express their feelings by words.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.

In Previously developed solution, they have to use coloured hand gloves for hand position recognition. Also, the old method uses traditional translators which take too much of time to process.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

In our device, there's an option called problem detection display in which our customer can able to see the type of problem occurs & solution will be displayed.

Focus on J&P, tap into BE, understand RRC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

By comparing normal people,Specially Abled people should depend on others and want to live their life independently like other people

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design

BEFORE: It is very difficult to convey the message to normal people.
AFTER: They overcome their reluctance to have communication with normal people.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

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.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

Advertise on online with influencers to test the product and promote it also on blog channels

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

On offline, we have our product experience stores where our customer can experience the product in real

Extract online & offline CH of BE

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

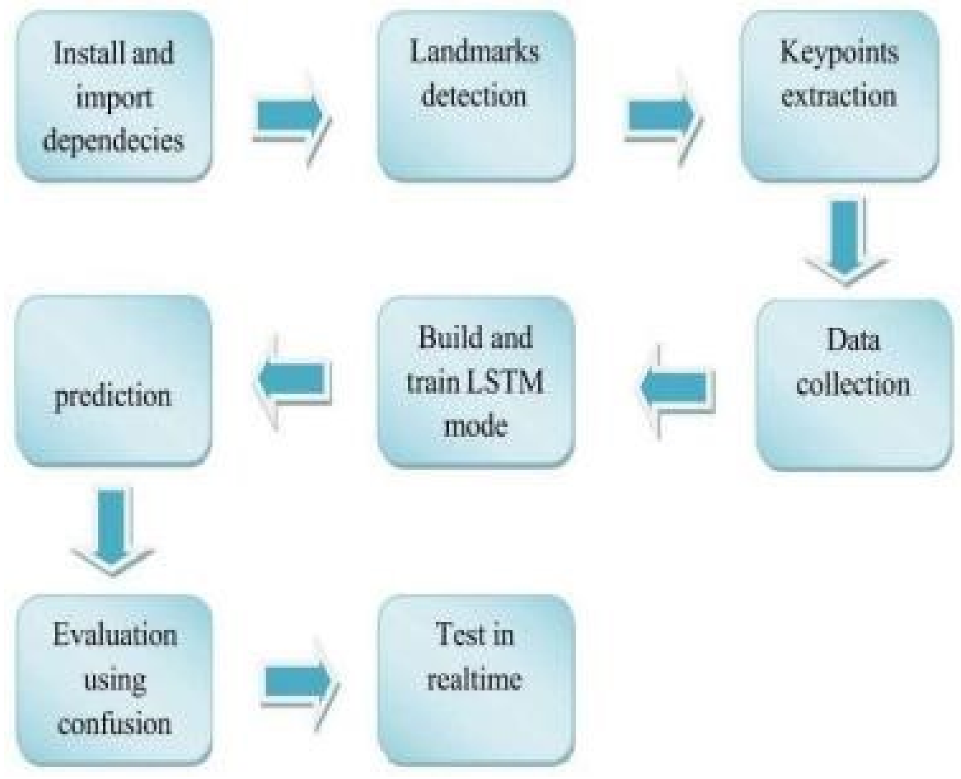
Non-functional Requirements:

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

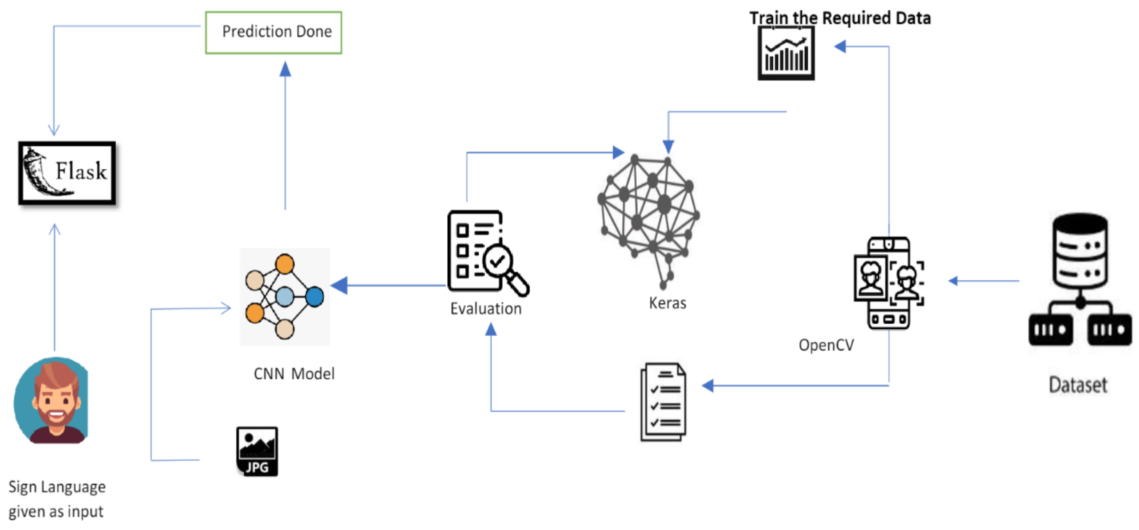
5.1 Data Flow Diagram



Install and import dependecies
Landmarks detection
Keypoints extraction
Data collection
Build and train LSTM mode
prediction
Evaluation using confusion
Test in realtime

5.2 Solution & Technical Architecture

Technology 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

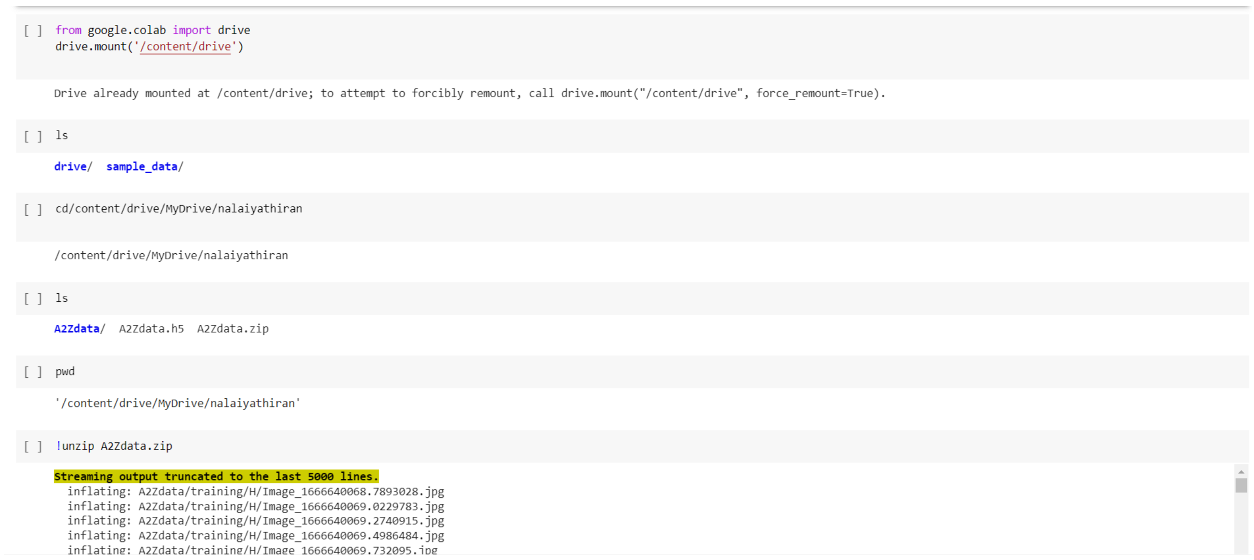
Milestone Activity Plan.

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
```

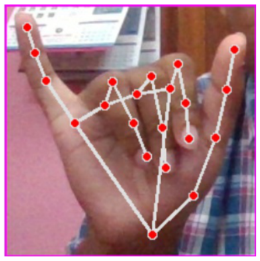
```
[ ] model=load_model('A2Z.h5')
```

```
[ ] pwd
```

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

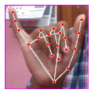
```
[ ] img=image.load_img(r'/content/drive/MyDrive/nalaiyathiran/A2Zdata/testing/Y/Image_1667328891.1069646.jpg')
```

```
[ ] 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}
```

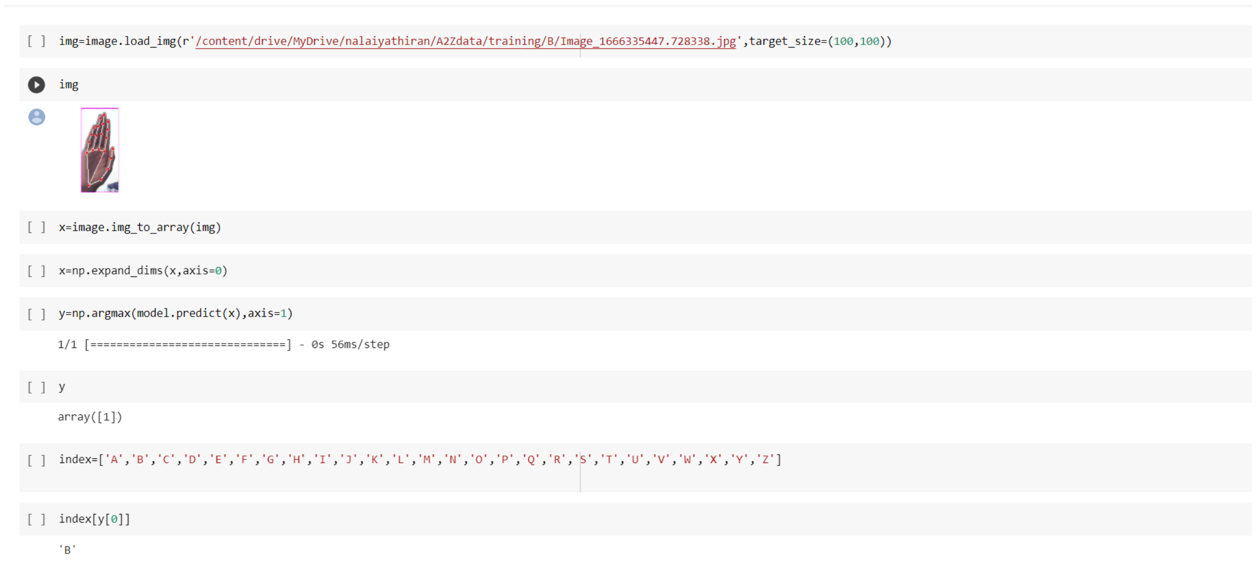
```
index=['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
```

```
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

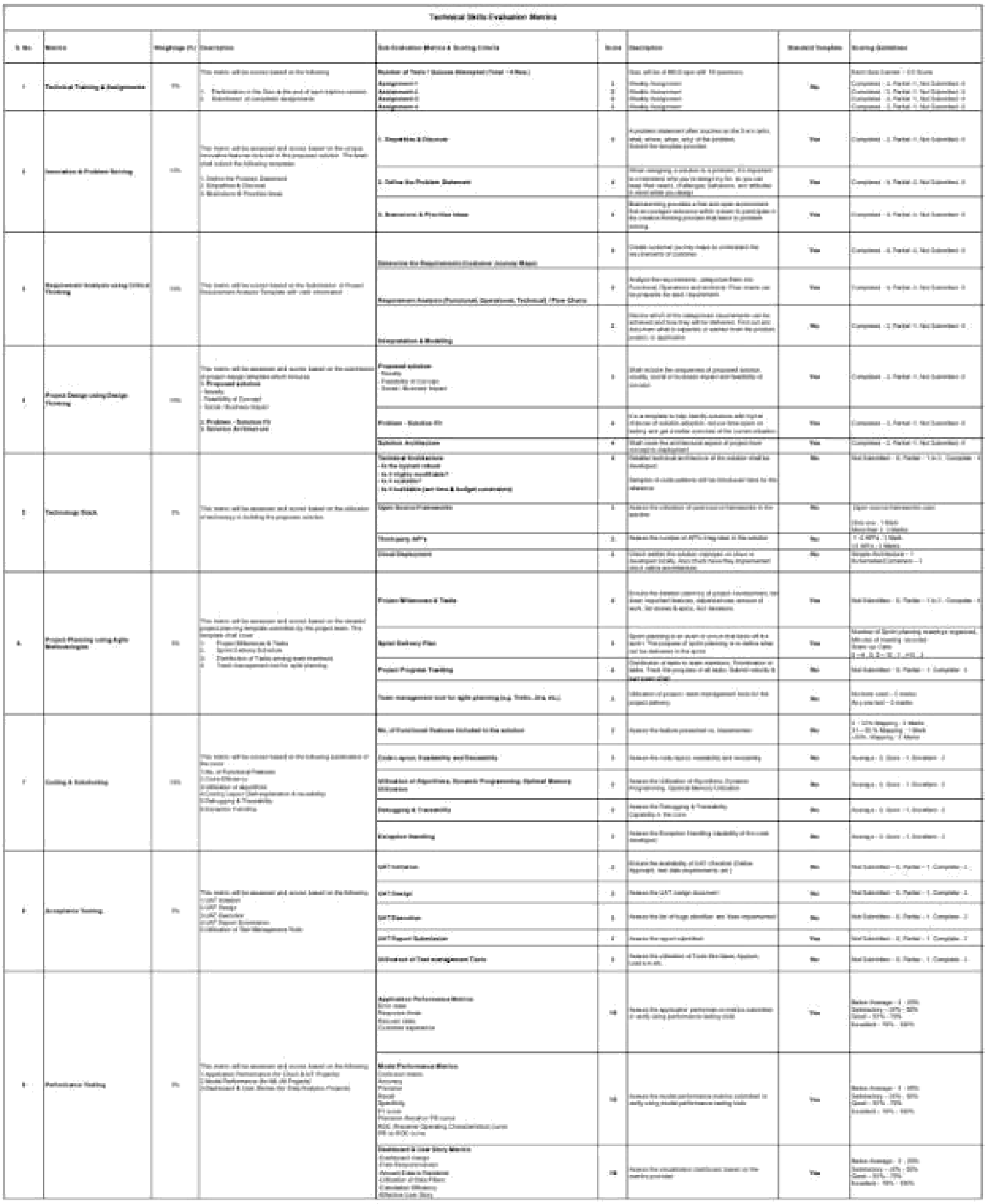
3. Test Case Analysis

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	[illegible]	[illegible]	Number of Tests / Quizzes Attempted (Total - 4 Nos.) Assignment-1 Assignment-2 Assignment-3 Assignment-4	[illegible]	[illegible]	No	[illegible]
2	Innovation & Problem Solving	[illegible]	[illegible]	1. Empathize & Discover	[illegible]	[illegible]	Yes	[illegible]
				2. Define the Problem Statement	[illegible]	[illegible]	Yes	[illegible]
				3. Brainstorm & Prioritize Ideas	[illegible]	[illegible]	Yes	[illegible]
3	Requirement Analysis using Critical Thinking	[illegible]	[illegible]	Determine the Requirements (Customer Journey Maps)	[illegible]	[illegible]	Yes	[illegible]
				Requirement Analysis (Functional, Operational, Technical) / Flow Charts	[illegible]	[illegible]	Yes	[illegible]
				Interpretation & Modelling	[illegible]	[illegible]	No	[illegible]
4	Project Design using Design Thinking	[illegible]	[illegible]	Proposed solution - Novelty - Feasibility of Concept - Social / Business Impact	[illegible]	[illegible]	Yes	[illegible]
				Problem - Solution Fit	[illegible]	[illegible]	Yes	[illegible]
				Solution Architecture	[illegible]	[illegible]	Yes	[illegible]
5	Technology Stack	[illegible]	[illegible]	Technical Architecture	[illegible]	[illegible]	No	[illegible]
				Open Source Frameworks	[illegible]	[illegible]	No	[illegible]
				Third-party API's	[illegible]	[illegible]	No	[illegible]
				Cloud Deployment	[illegible]	[illegible]	No	[illegible]
6	Project Planning using Agile Methodologies	[illegible]	[illegible]	Project Milestones & Tasks	[illegible]	[illegible]	Yes	[illegible]
				Sprint Delivery Plan	[illegible]	[illegible]	Yes	[illegible]
				Project Progress Tracking	[illegible]	[illegible]	No	[illegible]
				Team management tool for agile planning	[illegible]	[illegible]	No	[illegible]
7	Coding & Solutioning	[illegible]	[illegible]	No. of Functional Features Included in the solution	[illegible]	[illegible]	No	[illegible]
				Code Layout, Readability and Reusability	[illegible]	[illegible]	No	[illegible]
				Utilization of Algorithms, Dynamic Programming, Optimal Memory Utilization	[illegible]	[illegible]	No	[illegible]
				Debugging & Traceability	[illegible]	[illegible]	No	[illegible]
				Exception Handling	[illegible]	[illegible]	No	[illegible]
8	Acceptance Testing	[illegible]	[illegible]	UAT Initiation	[illegible]	[illegible]	No	[illegible]
				UAT Design	[illegible]	[illegible]	No	[illegible]
				UAT Execution	[illegible]	[illegible]	No	[illegible]
				UAT Report Submission	[illegible]	[illegible]	Yes	[illegible]
				Utilization of Test management Tools	[illegible]	[illegible]	No	[illegible]
9	Performance Testing	[illegible]	[illegible]	Application Performance Metrics	[illegible]	[illegible]	Yes	[illegible]
				Model Performance Metrics	[illegible]	[illegible]	Yes	[illegible]
				Dashboard & User Story Metrics	[illegible]	[illegible]	Yes	[illegible]

10.<u>ADVANTAGES & DISADVANTAGES</u>

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

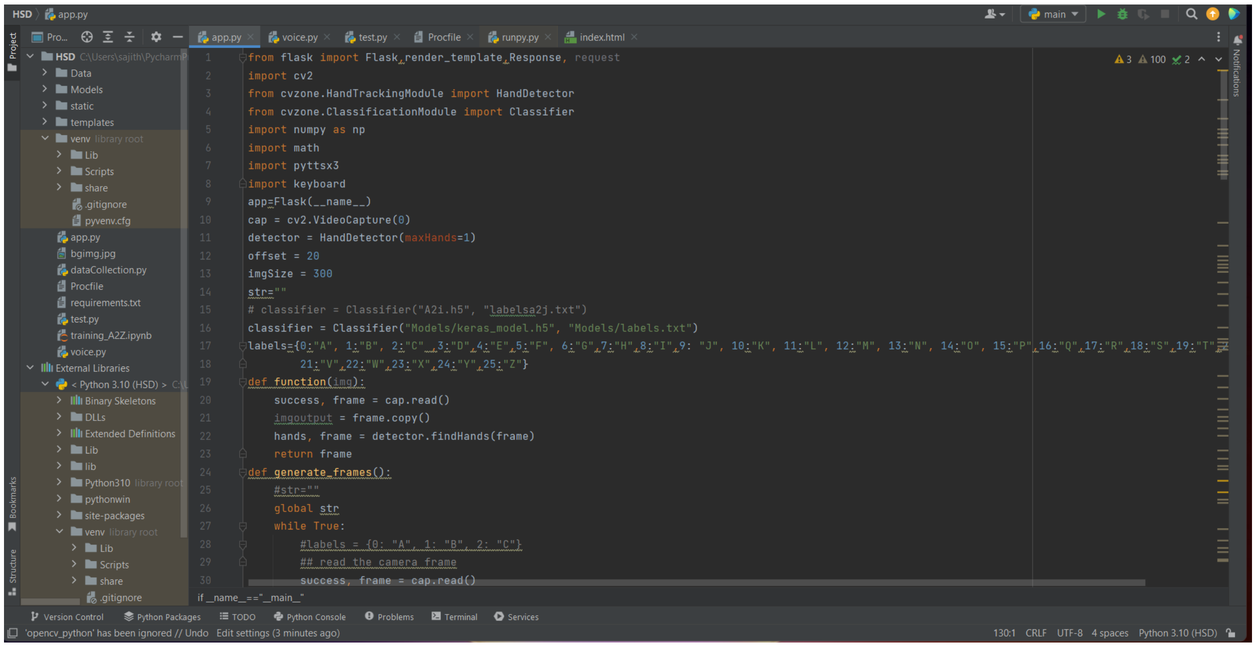
The speech-to-text and text-to-speech technologies helped those people who had difficulties in communicating or expressing their feelings to the normal people.

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

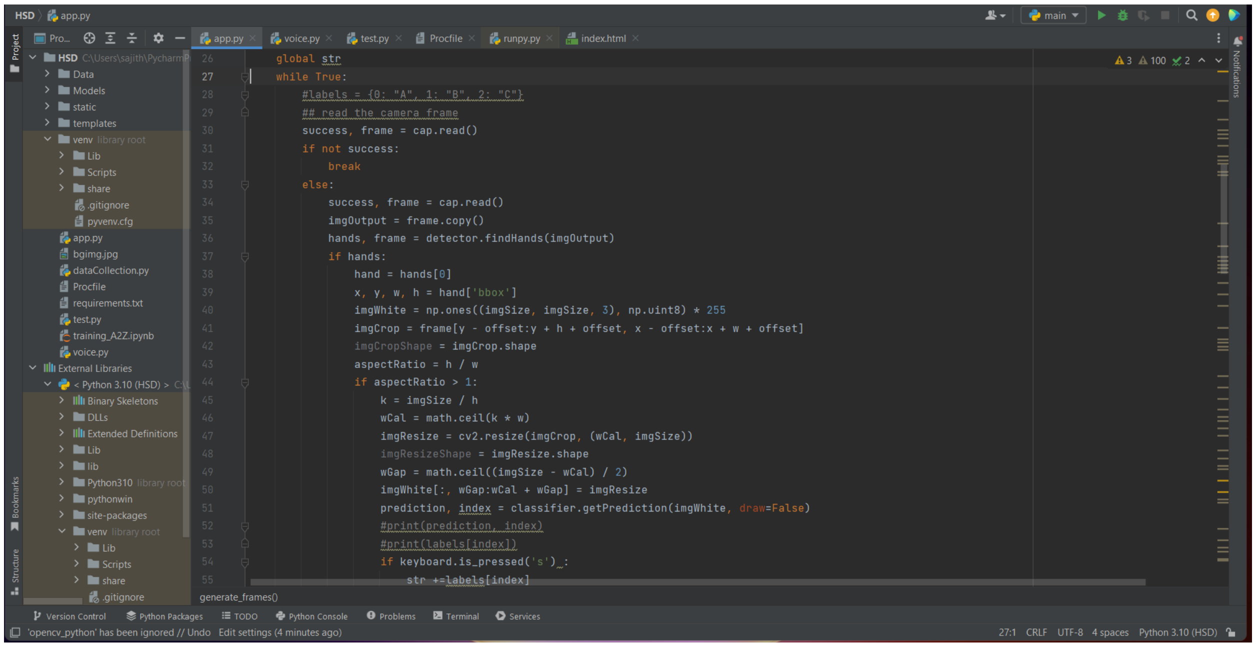
Source Code:



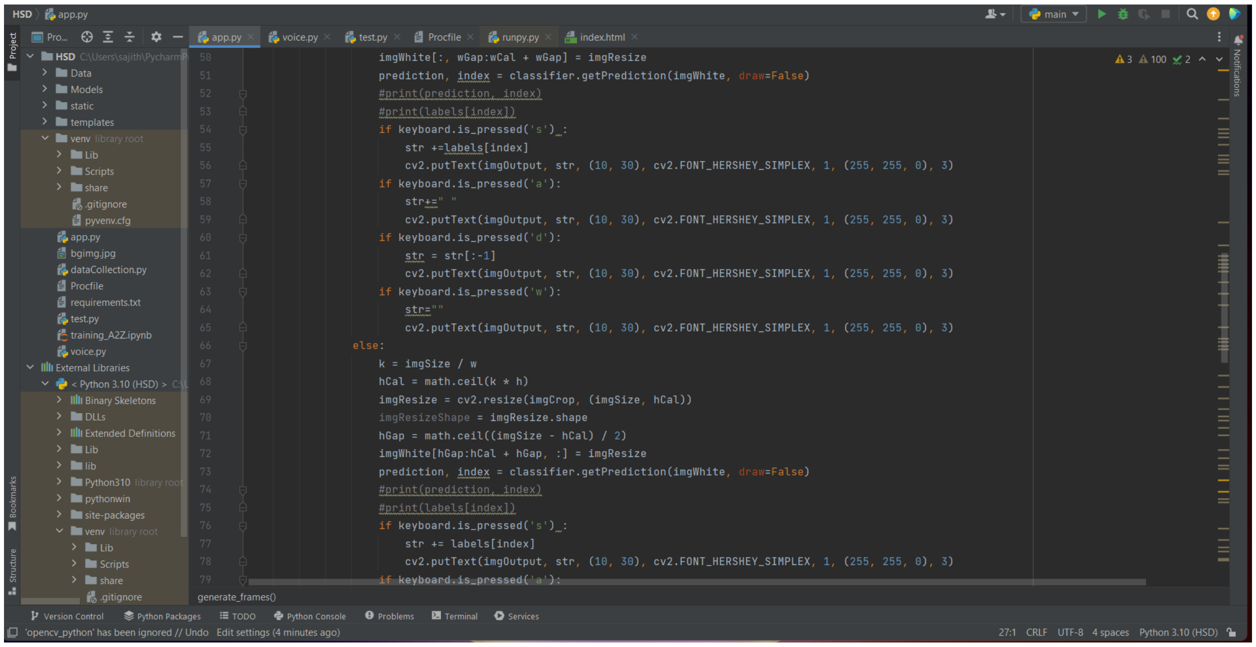
```python
from flask import Flask,render_template,Response, request
import cv2
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import numpy as np
import math
import pyttsx3
import keyboard
app=Flask(__name__)
cap = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1)
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__"

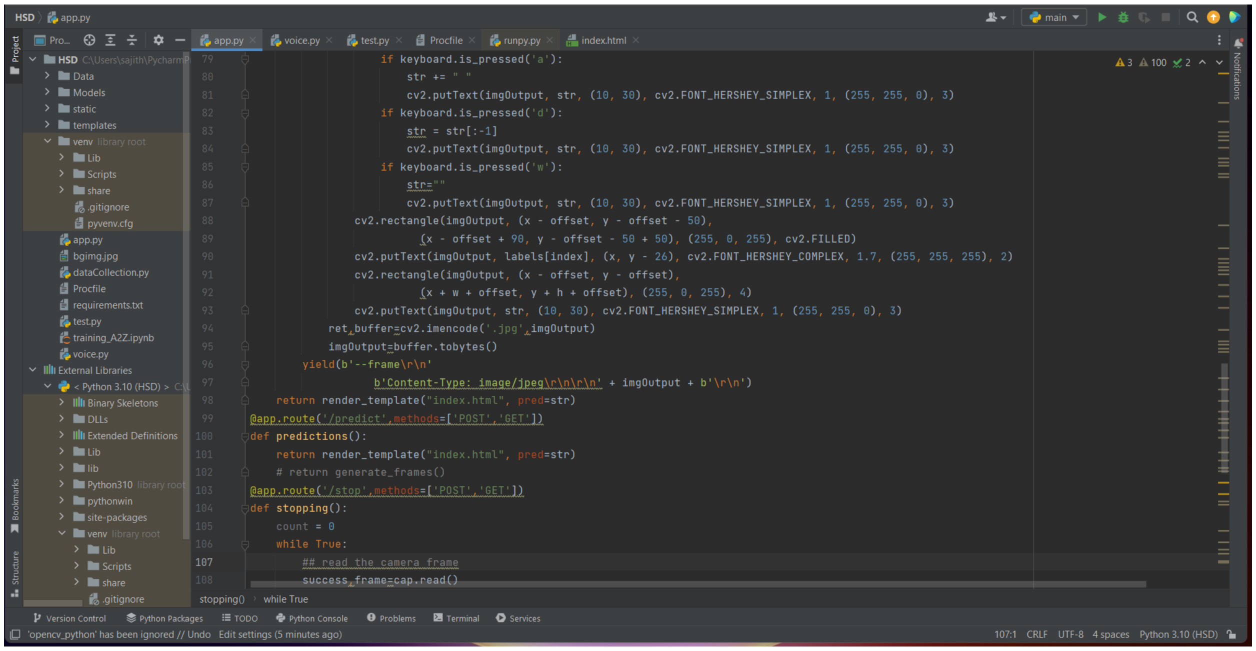
'opencv_python' has been ignored // Undo Edit settings (3 minutes ago)   130:1 CRLF UTF-8 4 spaces Python 3.10 (HSD)



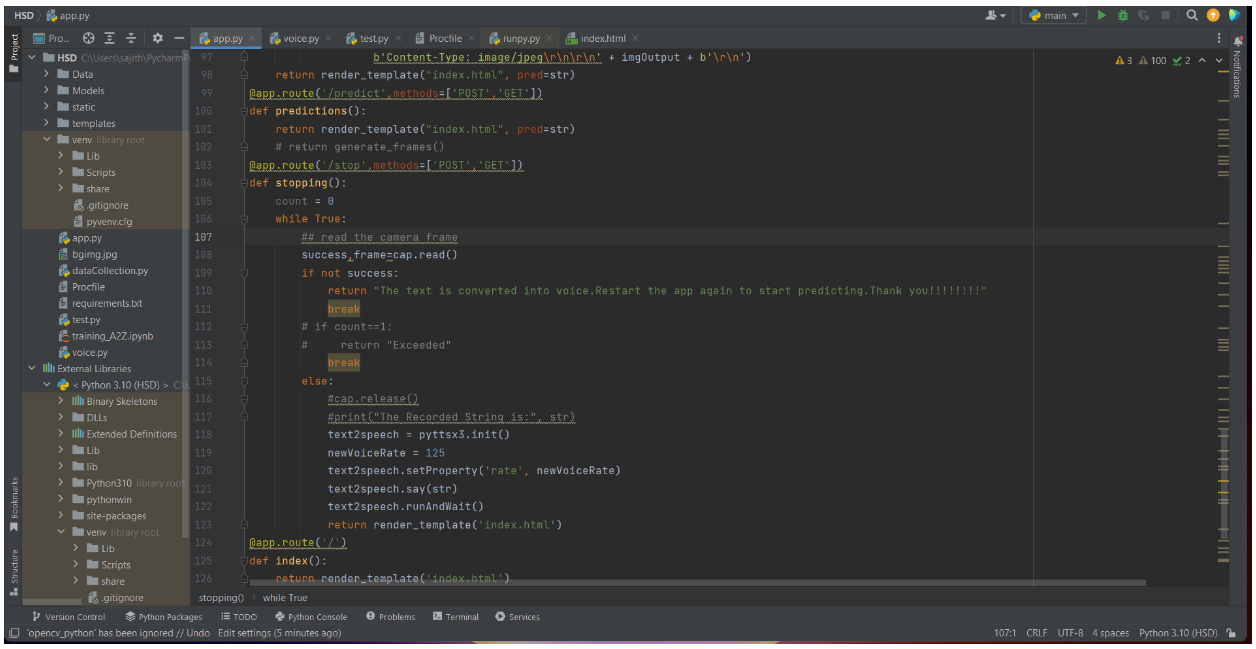
```python
    global str
    while True:
        #labels = {0: "A", 1: "B", 2: "C"}
        ## read the camera frame
        success, frame = cap.read()
        if not success:
            break
        else:
            success, frame = cap.read()
            imgOutput = frame.copy()
            hands, 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]
```



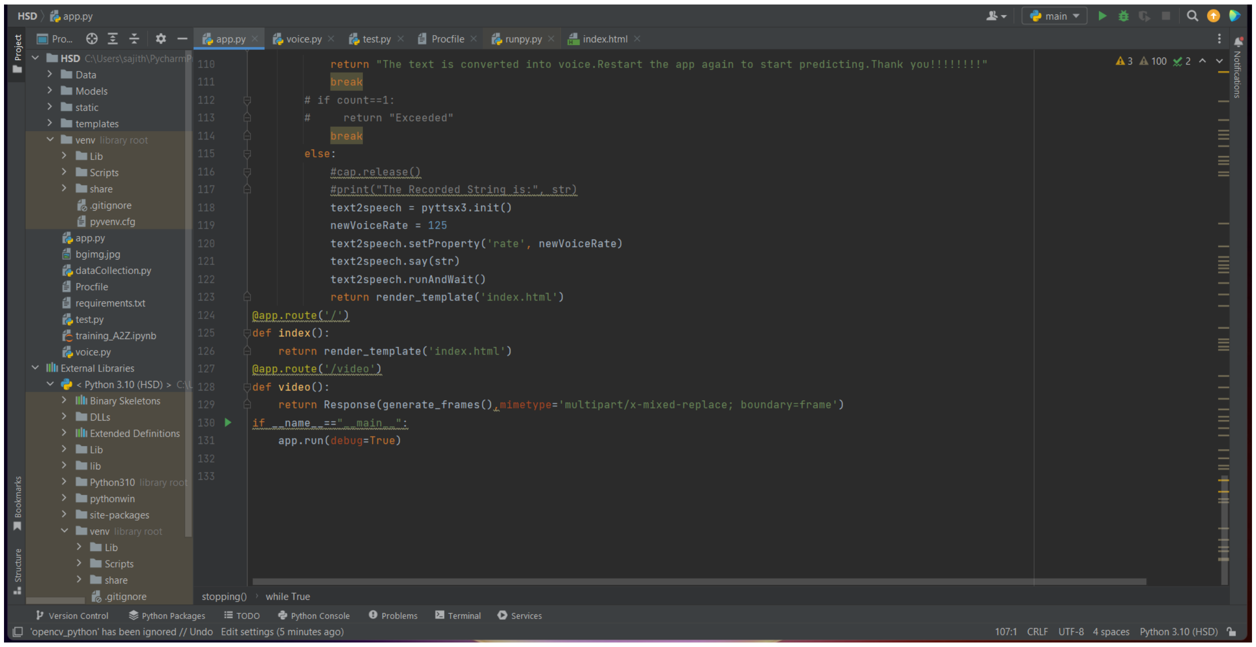
```python
                    imgWhite[:, wGap:wCal + wGap] = imgResize
                    prediction, index = classifier.getPrediction(imgWhite, draw=False)
                    #print(prediction, index)
                    #print(labels[index])
                    if keyboard.is_pressed('s') :
                        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'):
```



```python
                    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()
```



```python
                   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()
        if not success:
            return "The text is converted into voice.Restart the app again to start predicting.Thank you!!!!!!!!"
            break
        # if count==1:
        #     return "Exceeded"
            break
        else:
            #cap.release()
            #print("The Recorded String is:", str)
            text2speech = pyttsx3.init()
            newVoiceRate = 125
            text2speech.setProperty('rate', newVoiceRate)
            text2speech.say(str)
            text2speech.runAndWait()
            return render_template('index.html')
@app.route('/')
def index():
    return render_template('index.html')
```



```
                return "The text is converted into voice.Restart the app again to start predicting.Thank you!!!!!!!!"
                break
            # if count==1:
            #     return "Exceeded"
                break
            else:
                #cap.release()
                #print("The Recorded String is:", str)
                text2speech = pyttsx3.init()
                newVoiceRate = 125
                text2speech.setProperty('rate', newVoiceRate)
                text2speech.say(str)
                text2speech.runAndWait()
                return render_template('index.html')
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/video')
def video():
    return Response(generate_frames(),mimetype='multipart/x-mixed-replace; boundary=frame')
if __name__=="__main__":
    app.run(debug=True)
```

'opencv_python' has been ignored // Undo   Edit settings (5 minutes ago)

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-5609-1658811745