Real-Time Communication System Powered by AI for Specially Abled

TEAM ID : PNT2022TMID01231

- TEAM LEADER : R K Tharun Kumar
- TEAM MEMBER1: A D Lalith Kumar
- TEAM MEMBER2: Surya Xavier
- TEAM MEMBER3

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

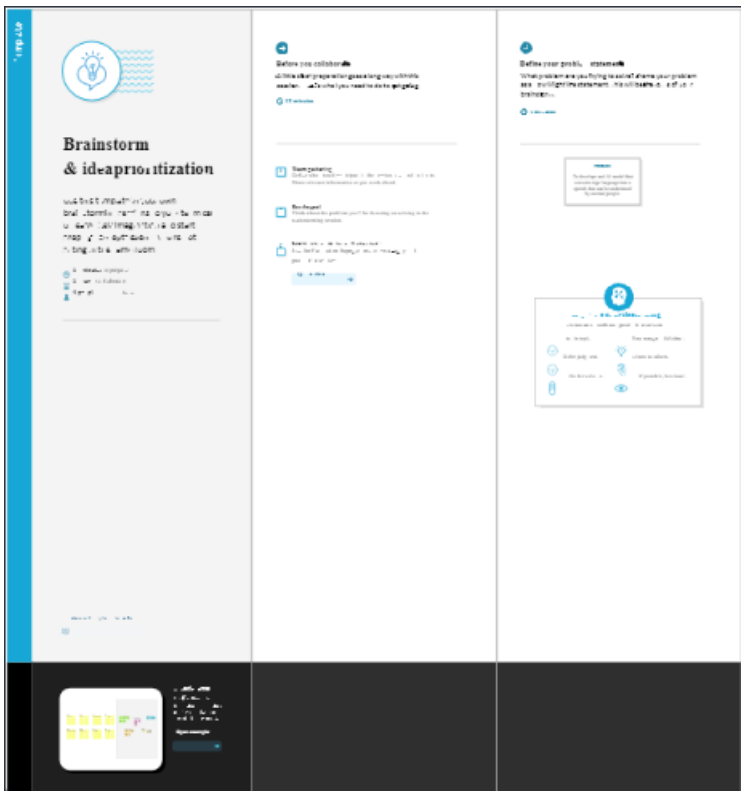
Only specially abled people are taught sign language and the common person is unaware its working causing a communication gap. Under emergency situations, it is even more difficult for specially abled people to get help. Non-Emergency normal environments can also be hard for them to navigate needing special assistance.

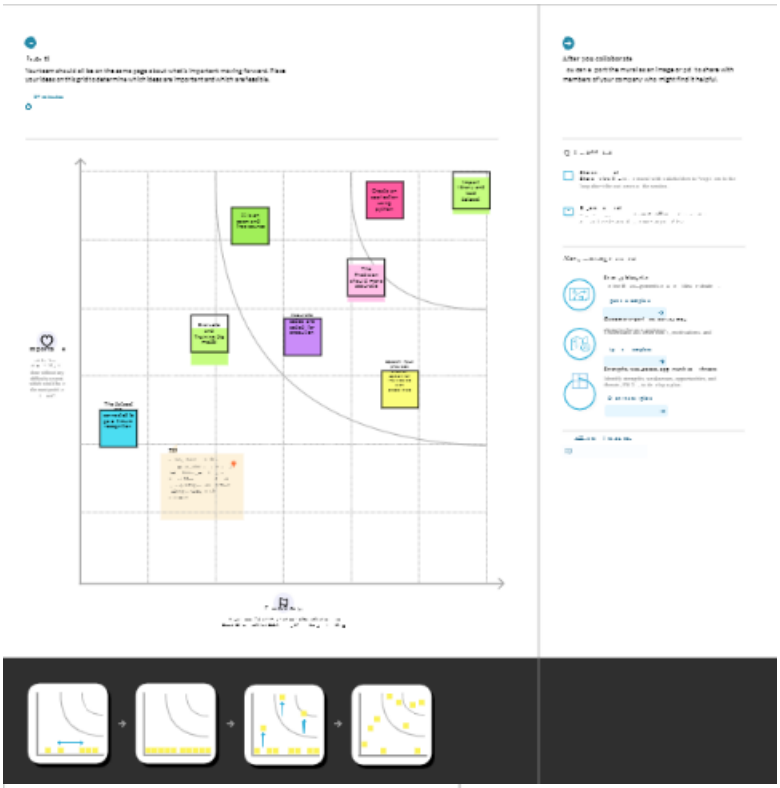
Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language.

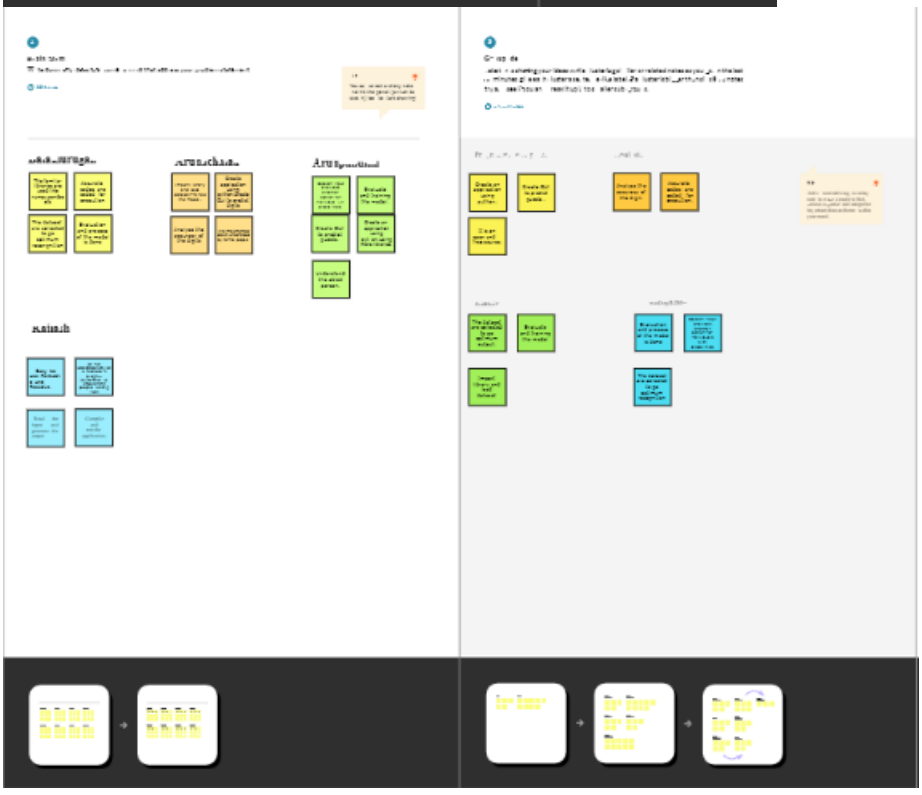
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming





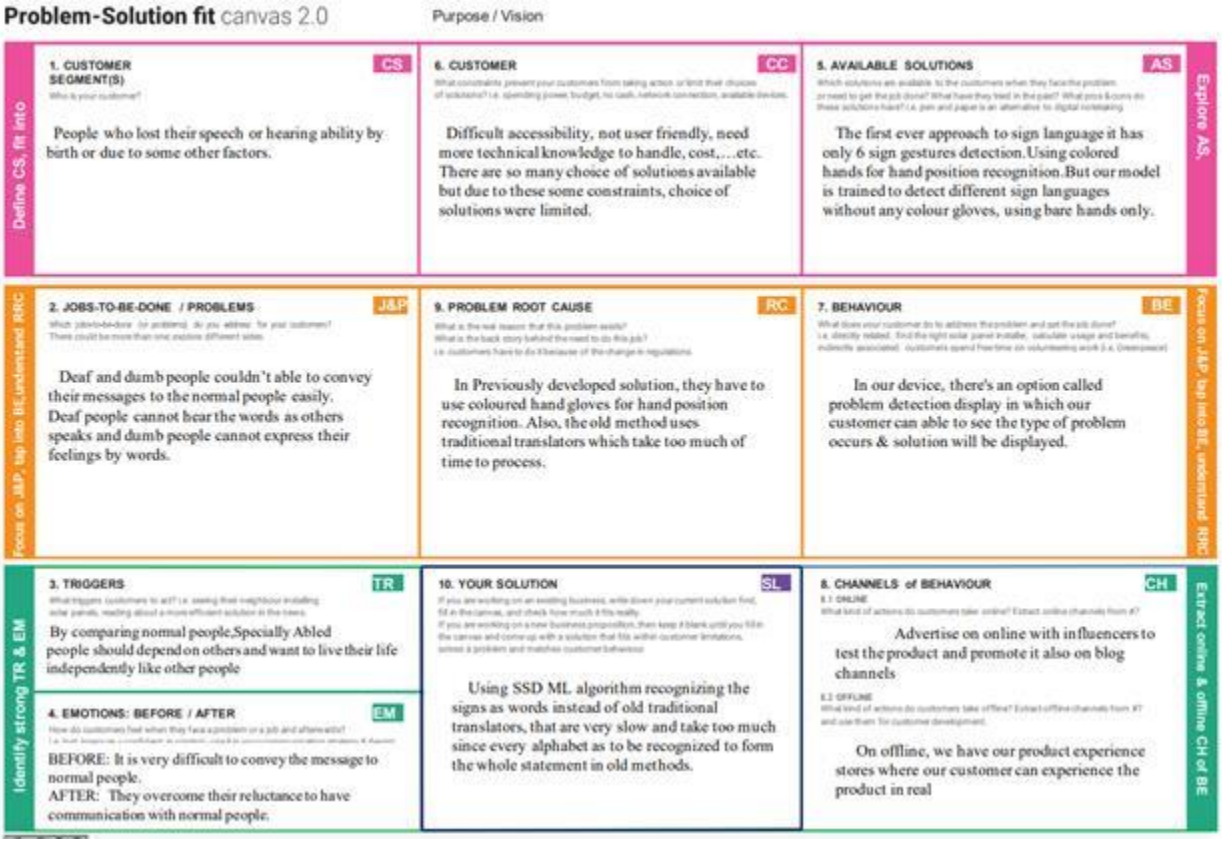


3.3 Proposed Solution

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

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Verification	The user should receive a verification e-mail which they have to confirm to complete the registration.
FR-4	Compliance to rules or laws	Terms and conditions, Privacy policy, End user licensing agreement.
FR-5	Authorization levels	There are two levels of authorization namely standard access level and advanced access level.
FR-6	Legal Requirements	Medical Certificate is produced

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The designed system is easy to use for specially abled persons as it is portable and platform independent.
NFR-2	**Security**	Converted information using signs into speech is accessed only by the user.
NFR-3	**Reliability**	System is tested with large number of data and Provides insight into issues.
NFR-4	**Performance**	Quick Launch time of application and faster in converting signs into speech
NFR-5	**Availability**	Provides automatic recovery and User access.
NFR-6	**Scalability**	Standard network condition the device should convert information within second.

4.2 Non-Functional requirements

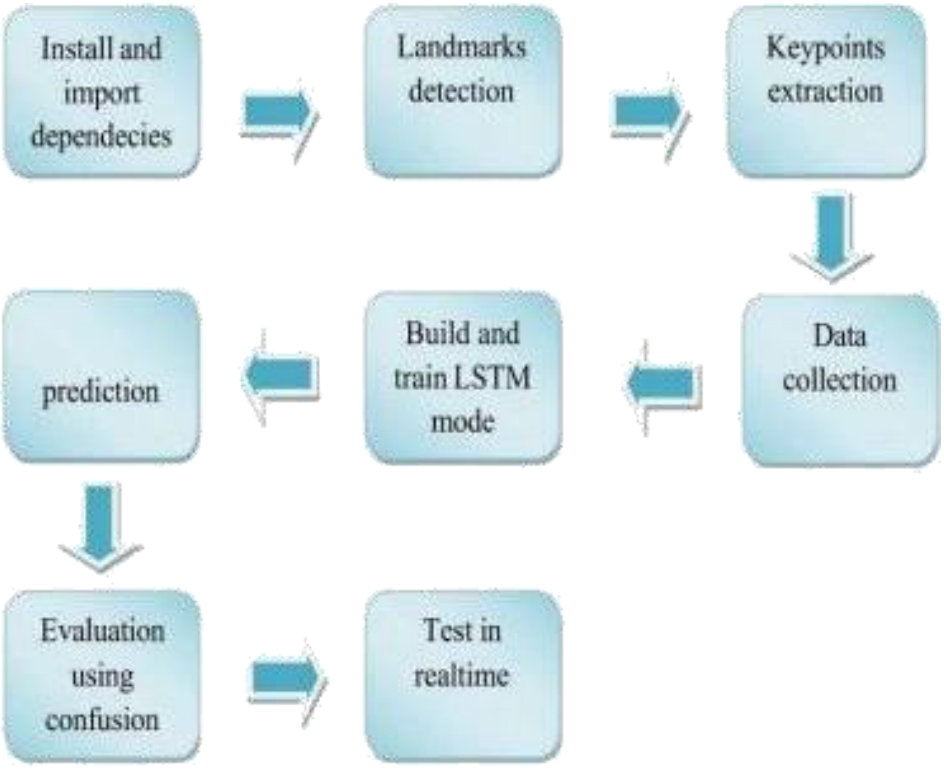
Non-functional Requirements:

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

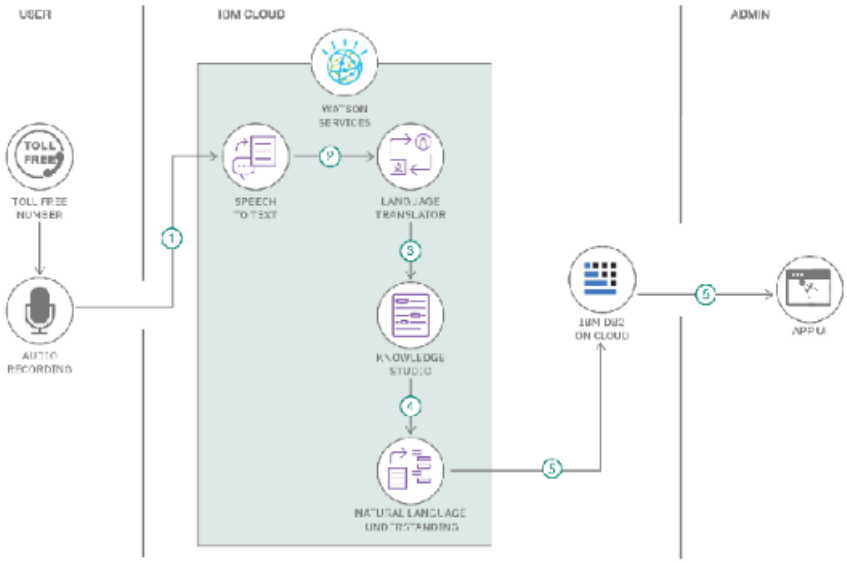
FR No.	Non-Functional Requirements	Description
NFR-1	**Usability**	The camera captures all expressions including facial expressions and hand gestures which can be easily used by all age groups. It can be used by deaf-mute people and their care takers.
NFR-2	**Security**	The system is more secure and 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	Team Members
Sprint – 1	Registration	USN – 1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High	R K Tharun Kumar
Sprint – 1	Authentication	USN – 2	As a user, I will receive OTP to confirm details.	2	High	R K Tharun Kumar
Sprint – 1	Registration	USN – 3	As a user, I will receive confirmation email once I have registered for the application.	1	Low	A D Lalith kumar
Sprint – 1	Login	USN – 4	As a user, I can log into the application by entering email & password.	2	High	A D Lalith kumar
Sprint – 2	Dashboard	USN – 5	As a user, I must have one place to explore all available features.	3	High	Surya Xavier
Sprint – 2	Login	USN – 6	As a user, If I forget my password, I must get an auto-generated password to reset my password.	2	Medium	Surya Xavier
Sprint – 3	Help	USN – 7	As a user, I must be able to reach out to the Support Team to get my issues resolved.	1	Low	R Arulmozhi Ganesh

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Milestone List:

Milestone Number	Milestone Name	Duration	Submission Dates
MN – 01	Ideation Phase	3 Weeks	17 September, 2022
MN – 02	Project Design Phase – I	2 Weeks	01 October, 2022
MN – 03	Project Design Phase – II	2 Weeks	15 October, 2022
MN – 04	Project Planning Phase	1 Week	22 October, 2022
MN – 05	Project Development Phase	3 Weeks	18 November, 2022
MN – 06	Pre-requisites	1 Week	30 September, 2022
MN – 07	Project Structure	1 Week	02 October, 2022
MN – 08	Data Collection	2 Days	04 October, 2022
MN – 09	Image Pre-processing	4 Days	08 October, 2022
MN – 10	Model Building	1 Week	19 October, 2022
MN – 11	Test the model	2 Days	20 October, 2022
MN – 12	Application Building	1 Week	22 October, 2022
MN – 13	Train CNN Model on IBM	2 Days	30 October, 2022

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint – 1	Registration	USN – 1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High	R K Tharun Kumar
Sprint – 1	Authentication	USN – 2	As a user, I will receive OTP to confirm details.	2	High	R K Tharun Kumar
Sprint – 1	Registration	USN – 3	As a user, I will receive confirmation email once I have registered for the application.	1	Low	A D Lalith kumar
Sprint – 1	Login	USN – 4	As a user, I can log into the application by entering email & password.	2	High	A D Lalith kumar
Sprint – 2	Dashboard	USN – 5	As a user, I must have one place to explore all available features.	3	High	Surya Xavier
Sprint – 2	Login	USN – 6	As a user, If I forget my password, I must get an auto-generated password to reset my password.	2	Medium	Surya Xavier
Sprint – 3	Help	USN – 7	As a user, I must be able to reach out to the Support Team to get my issues resolved.	1	Low	R Arulmozhi Ganesh

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

In [2]:
```
from keras.preprocessing.image import ImageDataGenerator
train_datagen= ImageDataGenerator(rescale= 1./255, shear_range = 0.2, zoom_range=0.2, horizontal_flip=True,vertical_flip=False)
test_datagen = ImageDataGenerator (rescale=1./255)
```

In [3]:
```
ls
```

```
drive/  sample_data/
```

In [4]:
```
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1./255)
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1./255,shear_range = 0.2, zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
```

In [5]:
```
cd /content/drive/MyDrive/IBM PROJECT/DATA COLLECTION
```

```
/content/drive/MyDrive/IBM PROJECT/DATA COLLECTION
```

In [ ]:
```
!unzip training_set.zip
```

```
Streaming output truncated to the last 5000 lines.
 extracting: training_set/G/1225.png
 extracting: training_set/G/1226.png
 extracting: training_set/G/1227.png
 extracting: training_set/G/1228.png
 extracting: training_set/G/1229.png
  inflating: training_set/G/123.png
 extracting: training_set/G/1230.png
 extracting: training_set/G/1231.png
 extracting: training_set/G/1232.png
  inflating: training_set/G/1233.png
  inflating: training_set/G/1234.png
  inflating: training_set/G/1235.png
  inflating: training_set/G/1236.png
  inflating: training_set/G/1237.png
```

```
[ ]  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
```

```
 inflating: test_set/I/86.png
 inflating: test_set/I/87.png
 inflating: test_set/I/88.png
 inflating: test_set/I/89.png
extracting: test_set/I/9.png
 inflating: test_set/I/90.png
 inflating: test_set/I/91.png
 inflating: test_set/I/92.png
 inflating: test_set/I/93.png
 inflating: test_set/I/94.png
 inflating: test_set/I/95.png
 inflating: test_set/I/96.png
 inflating: test_set/I/97.png
 inflating: test_set/I/98.png
 inflating: test_set/I/99.png
```

In [8]:
```
x_test = test_datagen.flow_from_directory('/content/drive/MyDrive/IBM PROJECT/DATA COLLECTION/test_set',target_size=(64,64),batch_size=300,class_mode=
```

```
Found 2250 images belonging to 9 classes.
```

In [9]:
```
print("Len x-train : ",len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train :  18
Len x-test :  8
```

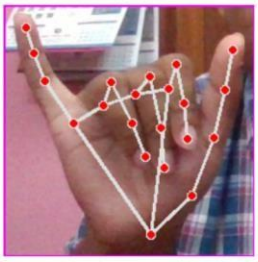
In [10]:
```
x_train.class_indices
```

Out[10]:
```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

In [11]:
```
x_test.class_indices
```

Out[11]:
```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

[ ]
```
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 & Modeling	[illegible]	[illegible]	No	[illegible]
4	Project Design using Design Thinking	[illegible]	[illegible]	Proposed solution	[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 (e.g. Trello, Jira, etc.)	[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	10	[illegible]	Yes	[illegible]
				Model Performance Metrics	10	[illegible]	Yes	[illegible]
				Dashboard & User Story Metrics	10	[illegible]	Yes	[illegible]

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

In [43]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [44]:
```
train_datagen = ImageDataGenerator(rescale = 1./255,shear_range=0.2,zoom_range= 0.2,horizontal_flip=True,vertical_flip=False)
```

In [45]:
```
test_datagen = ImageDataGenerator(rescale = 1./255)
```

In [46]:
```
import tensorflow as tf
import os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import IPython.display as display
from PIL import Image
import pathlib
```

In [47]:
```
from keras.preprocessing import image
# and use
# image.ImageDataGenerator()
# image.load_img()
```

Apply ImageDataGenerator Functionality To Train And Test set

In [48]:
```
from google.colab import drive
```

In [49]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
print("This dataset has been created and uploaded by IBM-TeamID-IBM-Project-PNT2022TMID00995")
```

```
This dataset has been created and uploaded by IBM-TeamID-IBM-Project-PNT2022TMID00995
```

In [50]:
```
x_train= train_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM PROJECT/DATA COLLECTION/training_set",target_size=(64,64),class_mode="categori
```

```
Found 15130 images belonging to 9 classes.
```

In [51]:
```
x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM PROJECT/DATA COLLECTION/test_set",target_size= (64,64),class_mode= "categorical
```

```
Found 2250 images belonging to 9 classes.
```

In [52]:
```
x_train.class_indices
```

Out[52]:
```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

In [53]:
```
x_test.class_indices
```

Out[53]:
```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

MODEL BUILDING

In [54]:
```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
```

In [55]:
```
model=Sequential()
```

In [56]:
```
model.add(Convolution2D(32,(3,3), input_shape=(64,64,1), activation = 'relu'))
```

In [57]:
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [58]:
```
model.add(Flatten())
```

In [59]:
```
model.add(Dense( units=512, activation='relu'))
```

In [60]:
```
model.add(Dense(units=9, activation='softmax'))
```

In [61]:
```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [62]:
```
model.save('Realtime.h5')
```

In [63]:
```
a=len(x_train)
b=len(x_test)
```

Length of training and testing data

In [64]:
```
print(a)
print(b)
```

```
316
47
```

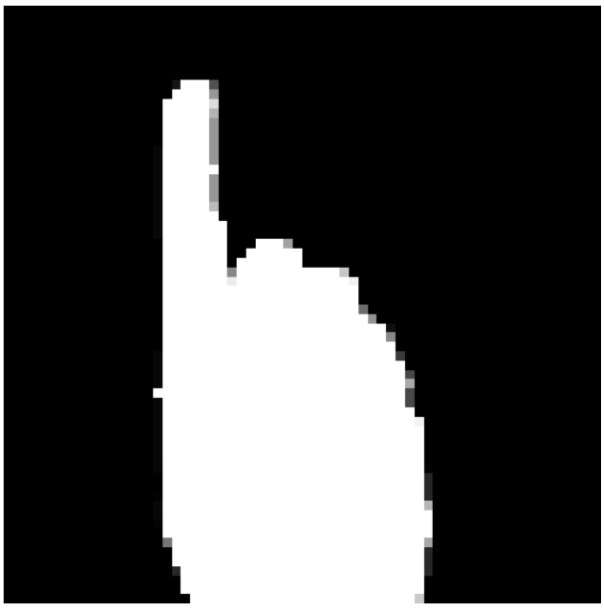
TEST THE MODEL

In [65]:
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import cv2
```


```
from tensorflow.keras.preprocessing import image
import numpy as np
import cv2
```


In [76]:
```
img = image.load_img('/content/drive/MyDrive/IBM PROJECT/DATA COLLECTION/test_set/D/101.png',target_size = (500,500))
img
```

Out[76]:



In [67]:
```
from skimage.transform import resize
arr=image.img_to_array(frame)
```

In [67]:
```
from skimage.transform import resize
arr=image.img_to_array(frame)
arr = resize(arr,(64,64,1))
arr = np.expand_dims(arr,axis=0)
pred=np.argmax(model.predict(arr))
op=['A','B','C','D','E','F','G','H','I']
print("THE PREDICTED LETTER IS ",op[pred])
```

```
1/1 [==============================] - 0s 65ms/step
THE PREDICTED LETTER IS  D
```

In [69]:
```
from skimage.transform import resize
def detect(frame):
  img=resize(frame,(64,64,1))
  img=np.expand_dims(img,axis=0)
  if(np.max(img)>1):
    prediction=model.predict(img)
    print(prediction)
    prediction=model.predict_classes(img)
    print(prediction)
```

In [70]:
```
arr= image.img_to_array(img)
```

In [71]:
```
frame=cv2.imread('/content/drive/MyDrive/IBM PROJECT/DATA COLLECTION/test_set/F/107.png')
data=detect(frame)
from google.colab.patches import cv2_imshow
cv2_imshow(frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [74]:
```
frame=cv2.imread('/content/drive/MyDrive/IBM PROJECT/DATA COLLECTION/test_set/A/102.png')
data=detect(frame)
```

In [74]:
```
frame=cv2.imread('/content/drive/MyDrive/IBM PROJECT/DATA COLLECTION/test_set/A/102.png')
data=detect(frame)
from google.colab.patches import cv2_imshow
cv2_imshow(frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [75]:
```
frame=cv2.imread('/content/drive/MyDrive/IBM PROJECT/DATA COLLECTION/test_set/D/108.png')
data=detect(frame)
from google.colab.patches import cv2_imshow
cv2_imshow(frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
print("THE PREDICTED LETTER IS ",op[pred])
```

```
THE PREDICTED LETTER IS  D
```

DEMO LINK:

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-2471-1658472163