ARTIFICIAL INTELLIGENCE

Real Time Communication System

Powered By AI

TEAM ID - PNT2022TMID22395

TEAM LEADER

Abinaya R - 113119UG04003

TEAM MEMBERS

Anusia M - 113119UG04014

Monika G - 113119UG04060

Durga S- 113119UG04301

VEL TECH MULTI TECH DR RANGARAJAN DR SAKUNTHALA ENGINEERING COLLEGE

CONTENT

1.Introduction:

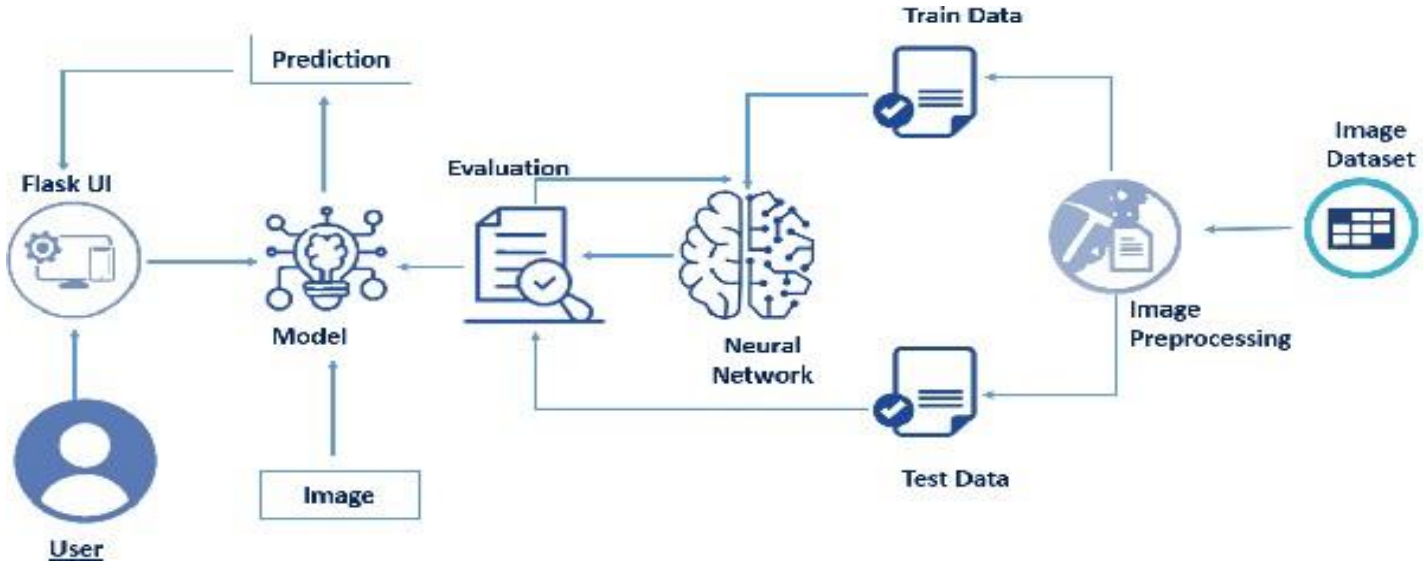
1.1 Project Overview:

Our project aims to provide an interface between deaf mute and normal people. Deaf mute people usually uses hand gesture as their sign language. But at times of emergency, it becomes difficult to convey their thoughts and messages. So, our motto is to create an interface between deaf mute and normal people.

1.2 Purpose:

We are using convolution neural network to provide real time communication system for specially abled. The sign language can be converted into speech and speech can be converted into sign language. By using this technology the barrier between deaf mute and normal people can be broken.

Block diagram:



2.Literature Survey

2.1 Literature Review:

SURVEY 1: AI Improving the Lives of Physically Disabled, HemaShree Madan

Physical disability is one of the significant concerns that hamper individuals to access the web easily. Most of the physically disabled people cannot use technology because of the limitation of accessibility tools and techniques. It is required that the websites should be made compliant with the requirements of every citizen in a country; that's why they should cater to the needs of the differently-abled citizens as well. Features have to be introduced in the websites so that they are easy to use, readily accessible, understandable, and convenient to everyone including best practices/standards and global innovation techniques. At times, accessibility is confused with providing solutions to disabled people, but the fact is accessibility is not only for differently-abled people, but it's also there for everyone. The matter is every person needs accessibility and uses it when in need.

SURVEY 2: "Artificial intelligence is like artificial god for Specailly Abled", Prem Mohan Newtech

Artificial Intelligence is not used to replace human beings rather It used to enhance their lives by helping us to do things that we are unable to do it on our own. These advancements have helped not just disabled individuals but also those who interact with them by making it easier to communicate without embarrassment or discomfort. The future is bright with the help of artificial intelligence.

SURVEY 3: " An AI software to communicate with deaf and mute in real time", Bhargav DV

The software, christened DnD Mate, does not only translate sign language into text and speech, but also translates speech into sign language, all in real time and as quick as the person speaks. Currently, there are no applications/software that facilitates a two-way communication channel. The software is based on a Deep Learning model and can work both offline and online. While in the offline mode, the deaf and mute person can communicate with you on the same device in real time; in the online mode, you can converse sitting in far off places as well, just like you talk to anyone over a video call

SURVEY 4: "voice-based device to assist deaf and mute",Dhaya Sindhu.

This device increases the listening experience for deaf people by converting the sign language into audio output. They've designed this device to be worn by the user on their face like sunglasses. It also has earphones attached to it which sends audio messages. The main features that have been incorporated in this device are the sensors, microcontrollers, LCD module, SD card, and audio amplifier. A sensor is needed in this device to pick up body language, especially the hand gestures. This device functions similar to a translation system, with the goal of converting sign language into audio.

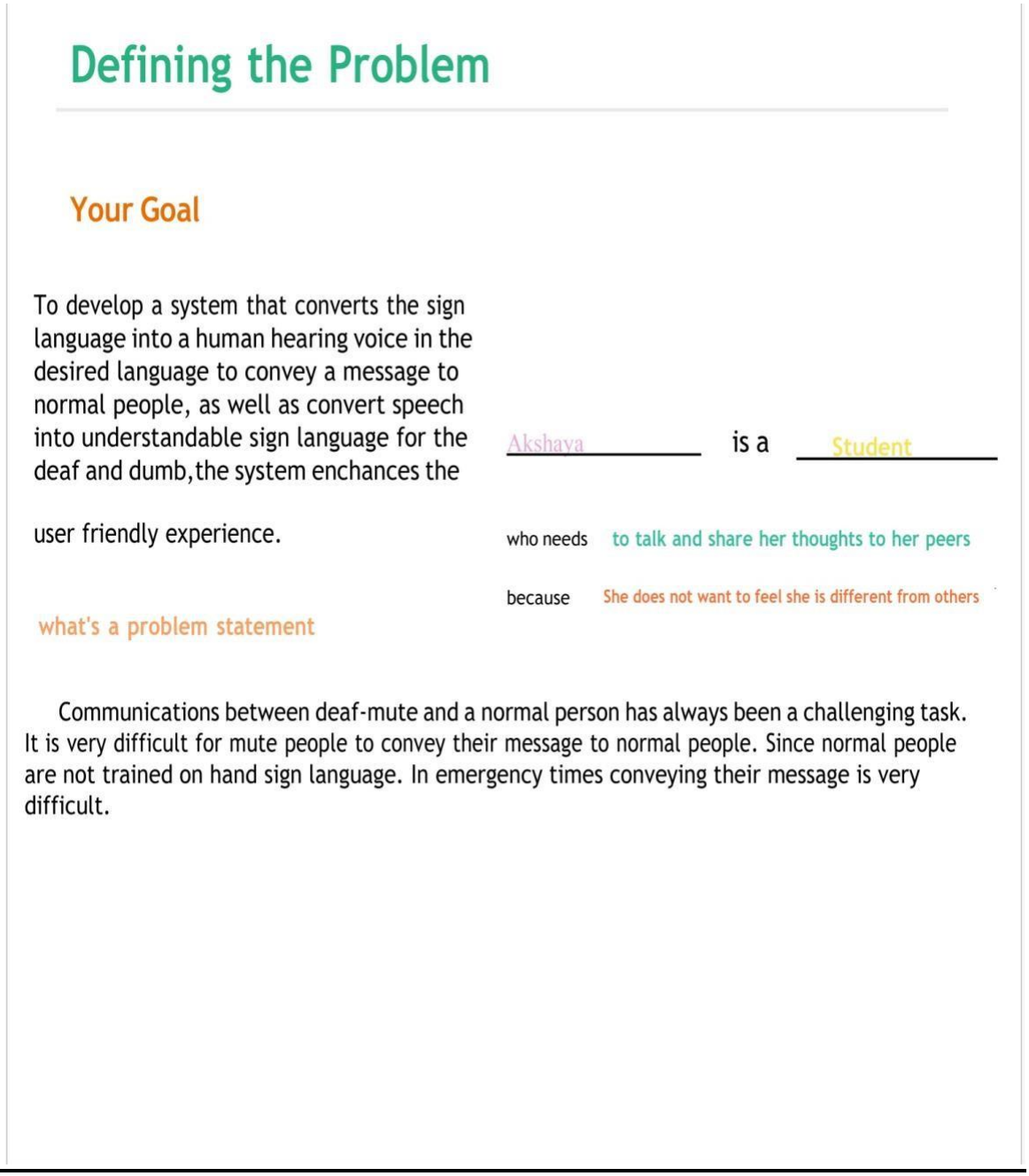
SURVEY 5: "Innovative study of an AI voice based smart Device to assist deaf people in understanding and responding to their body language", Lakshmi sri surya.

People who have hearing impairments have difficulty communicating with those who do not have hearing issues if they do not have access to a translator [1]. This is why the deaf community will benefit greatly from a technology that understands sign language especially hand gestures. Even though mobile technology is rapidly evolving and becoming incredible, there has been technological advancement and development for artificial intelligence voice based smart dev ices that can assist deaf people in understanding and responding to their body language.

2.2 Existing problem:

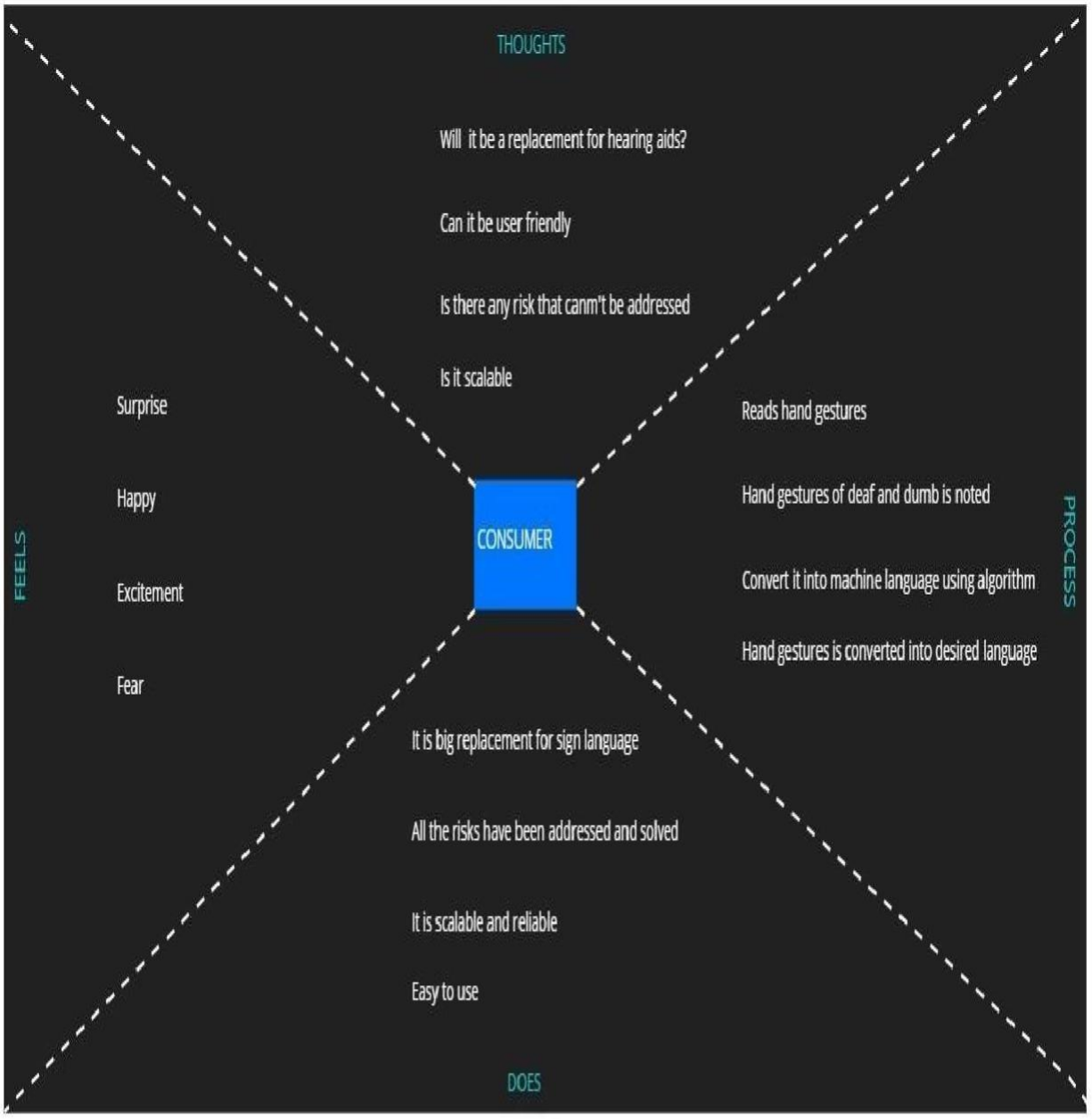
Deaf mute people find it difficult to share their thoughts and it becomes challenging during emergency. Because normal people aren't trained with hand gesture or sign language. So, it is necessary to provide an interface between deaf mute and normal people.

2.3 Problem Statement Definition:



3.Ideation Phase And Proposed Solution:

3.1 Empathy map:



3.2 Ideation and Brainstorming:

IMPROVING USER EXPERIENCE :-

Getting frequent feedbacks

Reducing the cost

Analysing the risk factors beforehand

USER HELP :-

Creating chat bot

Creating tutorials

Attaching user manual along with the product

FEATURES :-

Creating the clear voice

Can be available in markets easily

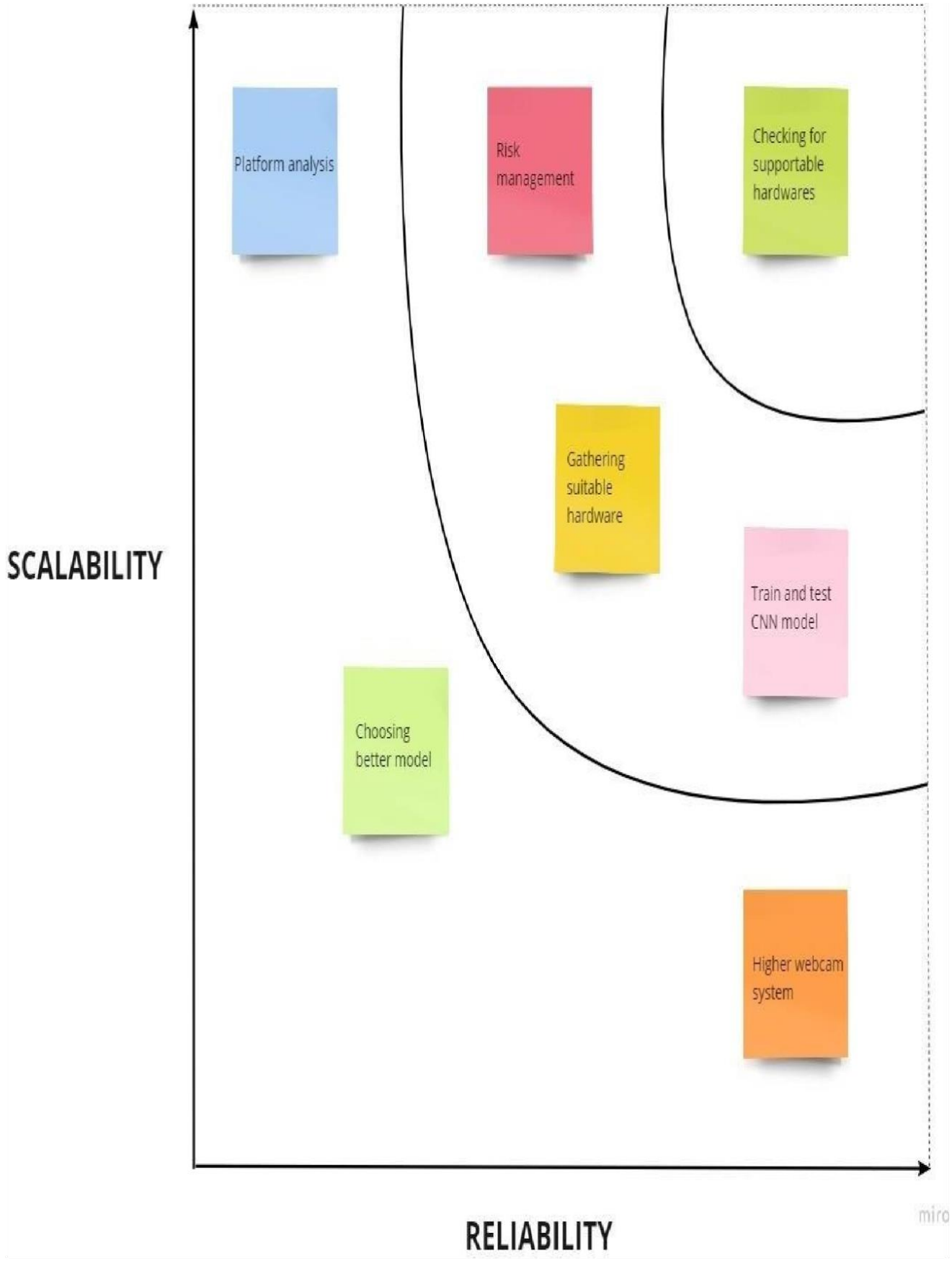
Creating a notepad

USING IMAGE PROCESSING :-

Converts hand gesture

Easily understandable by people

User friendly


Platform analysis
Risk management
Checking for supportable hardwares
Gathering suitable hardware
Train and test CNN model
Choosing better model
Higher webcam system
SCALABILITY
RELIABILITY
miro

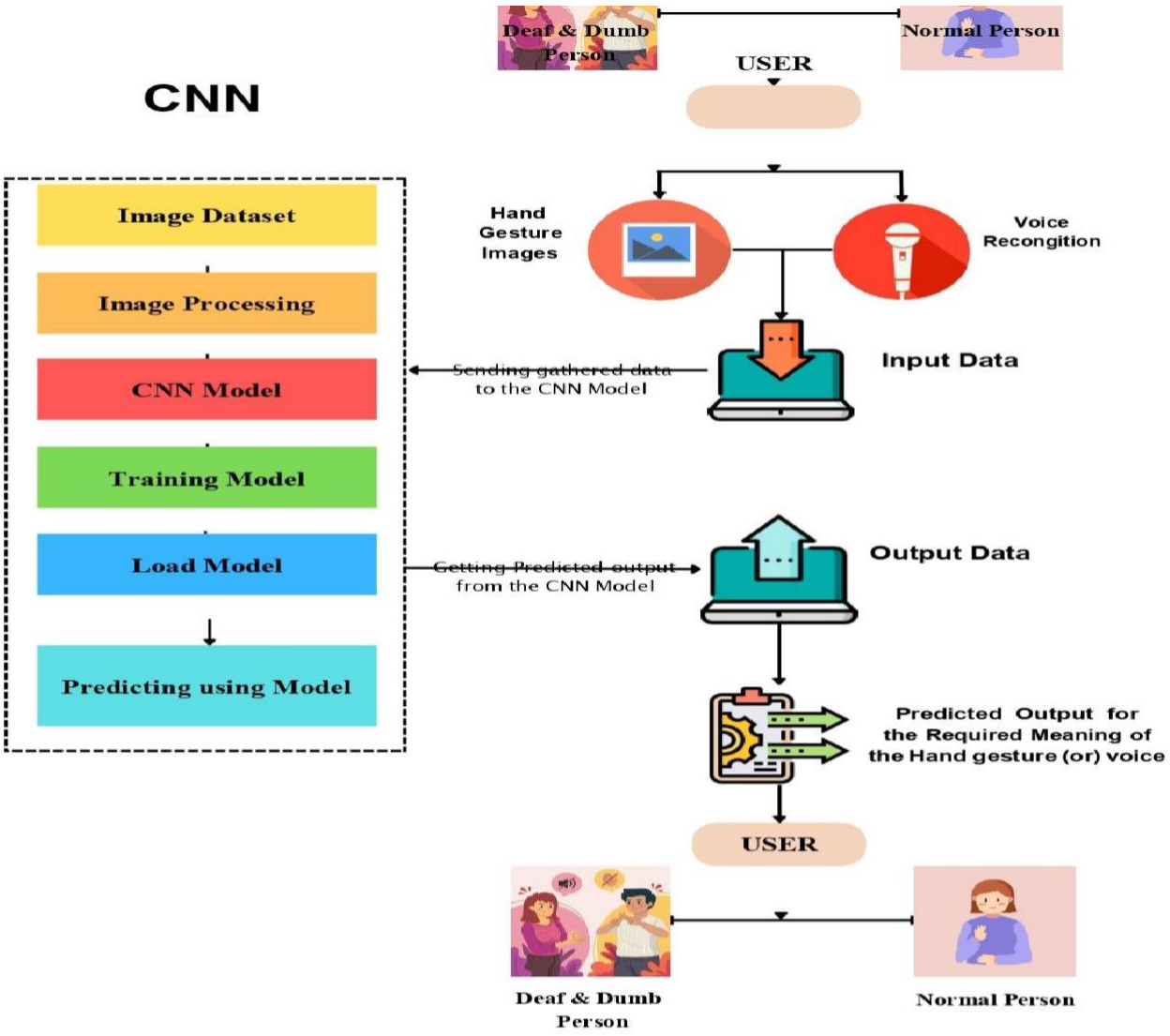
3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• To help specially abled people to communicate effectively to normal people. • To help them live their life without any greater efforts • To provide communication between normal people and specially abled people.
2.	Idea / Solution description	• Converting hand gestures into voice translation • Converts sign language into human hearing voice in desired language • An app is built to train different hand gesture
3.	Novelty / Uniqueness	• Captures hand gestures • Converts hand gesture into desirable language • This will be conveyed to Normal people
4.	Social Impact / Customer Satisfaction	• The main aim of the project is to provide interference and break the barrier between specially abled people and normal people • To help specially helped people convey their messages without any great efforts. • The speech is the utmost output of the project.

5.	Business Model (Revenue Model)	• Uses to have access to this interface changes specially abled people life better one • By using mobile application , tracking the number of people using this technology currently. • Charging them a little to enhance the system Performance
6.	Scalability of the Solution	• Can be easily used by people with reduced mobility and control everything at home • It can be a great alternative for hearing aids and sign language • Everyone can have access to this technology including people with disability.

1. To design and develop a system which lowers the communication gap between speech hearing impaired and normal world.
2. To build a communication system that enables communications between deaf-dumbperson and a normal person.

3.4 Problem Solution Fit:



4.Requirement Analysis:

4.1 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 Form • Registration through Gmail • Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP

4.2 Non-functional Requirements:

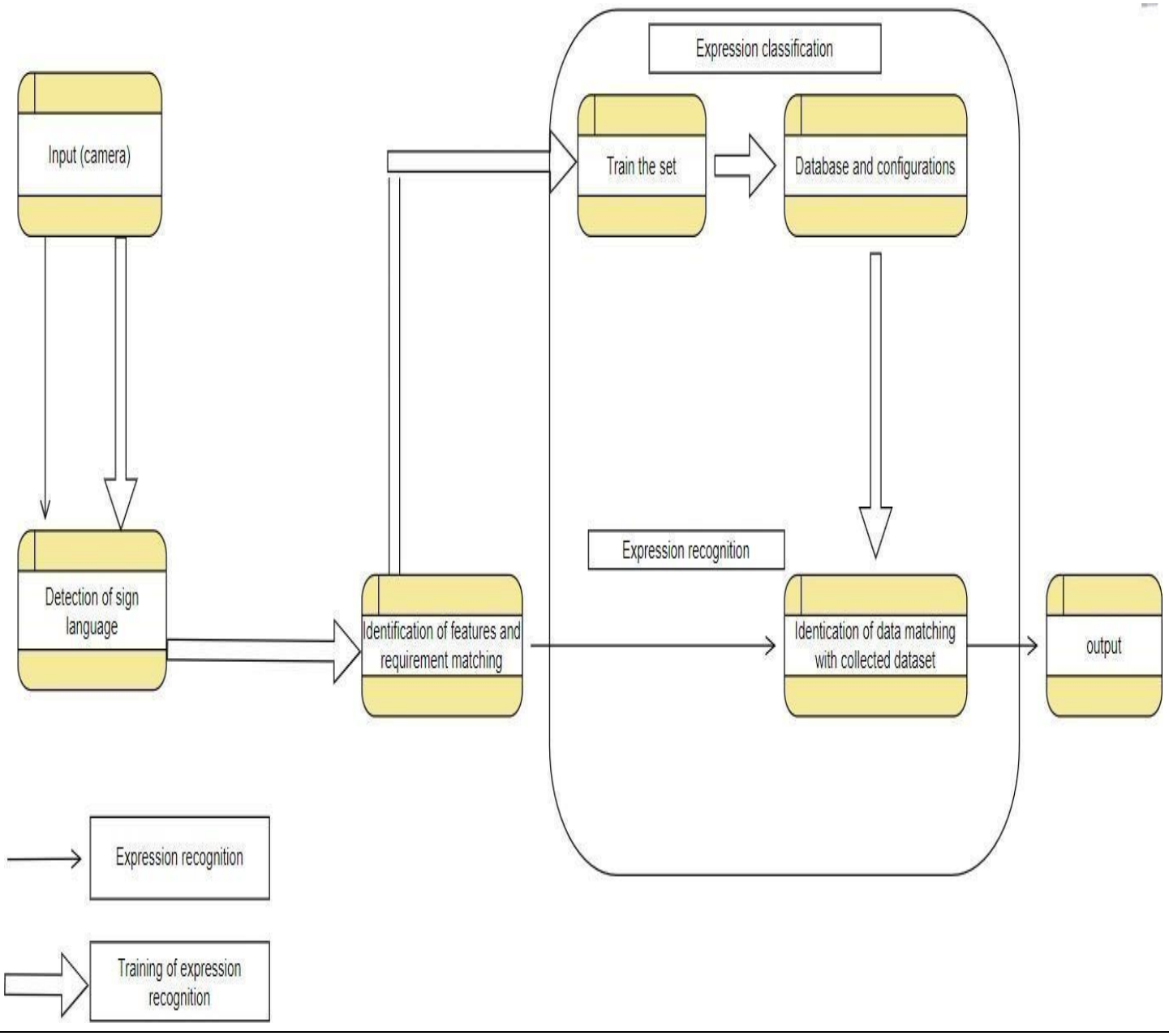
Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The product must be user friendly and it must beable to accessed by everyone.
NFR-2	**Security**	The customers details must be safe and secured and not to be sold for third party
NFR-3	**Reliability**	The product being developed should be reliable andto be initialized without any issue
NFR-4	**Performance**	The accuracy must be improved and there must be regular updates to improve the performance.
NFR-5	**Availability**	The customer support must be available 24/7 andqueries must be resolved time to time.
NFR-6	**Scalability**	80-85% is the percentage of scalability

5.Project Design Phase:

5.1 Data Flow Diagram:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 User Stories

User Stories

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I can register for the application through Gmail	I can access my account / dashboard	High	Sprint-1
	Confirmation	USN-3	As a user, I will receive confirmation emailonce I have registered for the application	I receive confirmationemail & click confirm	High	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password	I can enter into the loginby id and password	High	Sprint-1
	Data input	USN-5	User will be giving the input, the camera as speech or signs	I can give the input to thesystem	High	Sprint-4
		USN-6	The system will take the input for the testing	They will accept theinputfor the testing	High	Sprint-2
	Data verification	USN-7	It will verify with the data base that will matchwith the input	Configuration of the input	High	Sprint-2
		USN-8	Identification of the input and convert into the text if the input is signs or as signs	Identification of the input and creating output	High	Sprint-3
	Output Display	USN-9	Display the output on the screen for the user	Display of the output	High	Sprint-2

5.3 Solution and technical architecture:

S.No	Component	Description	Technology
1.	User Interface	The user interface is the point of human computer interaction and communication in device	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Converting speech into sign language	Java / Python
3.	Application Logic-2	Converting sign language to speech	IBM Watson STT service
4.	Application Logic-3	Converting speech to readable content	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, Rational database etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloud etc.
7.	File Storage	Methodology used to organize and store data on a computer hard drive	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API	Defines communication between normal people and deaf people	IBM Weather API, etc.
9.	Machine Learning Model	Training	Object Recognition Model, etc.

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use offirewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Devops
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Conferencing technology
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	NLP

6.Project Planning and Scheduling:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Member
Sprint-1	Data Collection	USN-1	Collect Dataset .	9	High	Abinaya R Anusia M
Sprint-1		USN-2	Image preprocessing	8	Medium	Abinaya R Anusia M
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	10	High	Monika G Durga S
Sprint-2		USN-4	Training the image classification modelusing CNN	7	Medium	Abinaya R Anusia M
Sprint-3	Training and Testing	USN-5	Training the model and testing the model's performance	9	High	Abinaya R Monika G
Sprint-4	Implementation ofthe application	USN-6	Converting the input sign language images into English alphabets	8	Medium	Anusia M Durga S

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	04 Nov 2022	5	04 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	11 Nov 2022	7	11 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	18 Nov 2022	5	18 Nov 2022

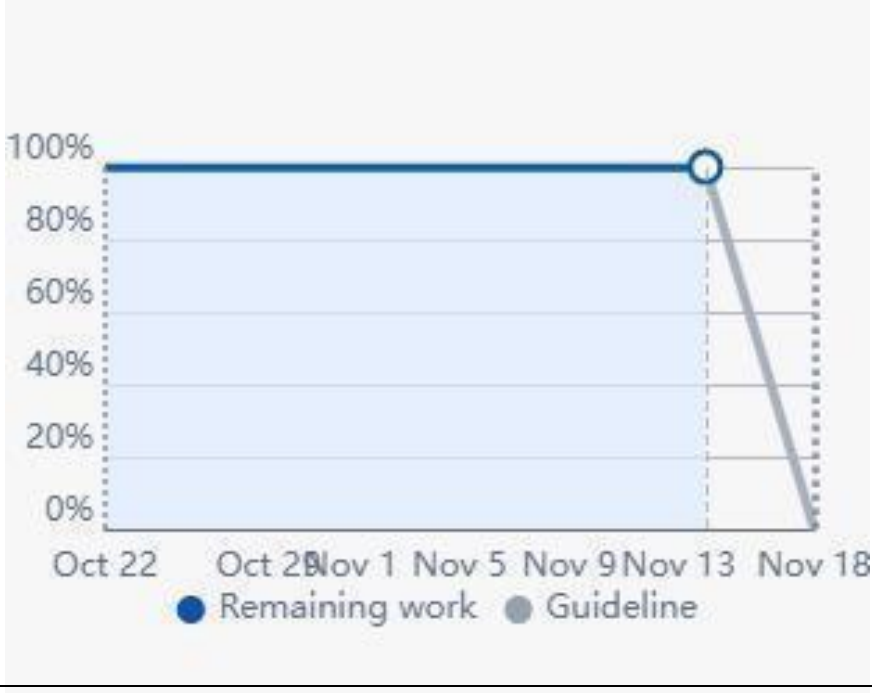
Velocity

$$AV = \frac{sprint\ duration}{velocity}$$

AV= 6/10 = 0.6

Burndown chart:

SPRINT BURNDOWN CHART:



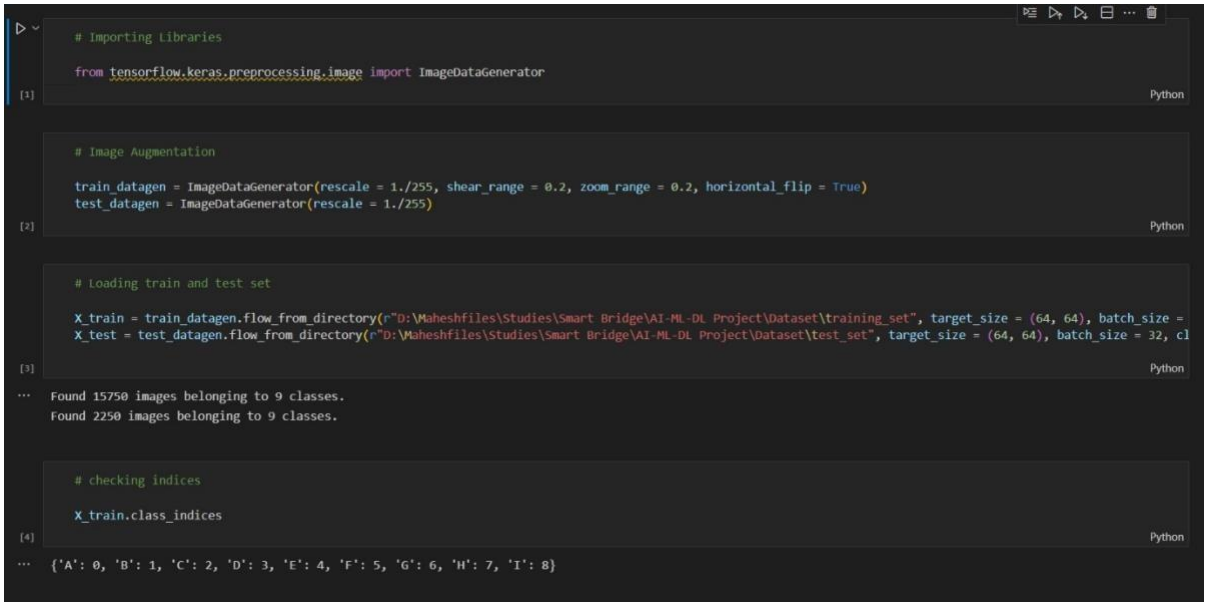
7.Coding and solutioning:

7.1 Features:

- Real time communication will break the barrier that is formed between Deaf mute and normal people
- Sign language is converted into text and text is converted into sign language
- With just the tap button, Communication between deaf mute and normal people will be broken.

7.2 Code:

Training the dataset



```
# Importing Libraries

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

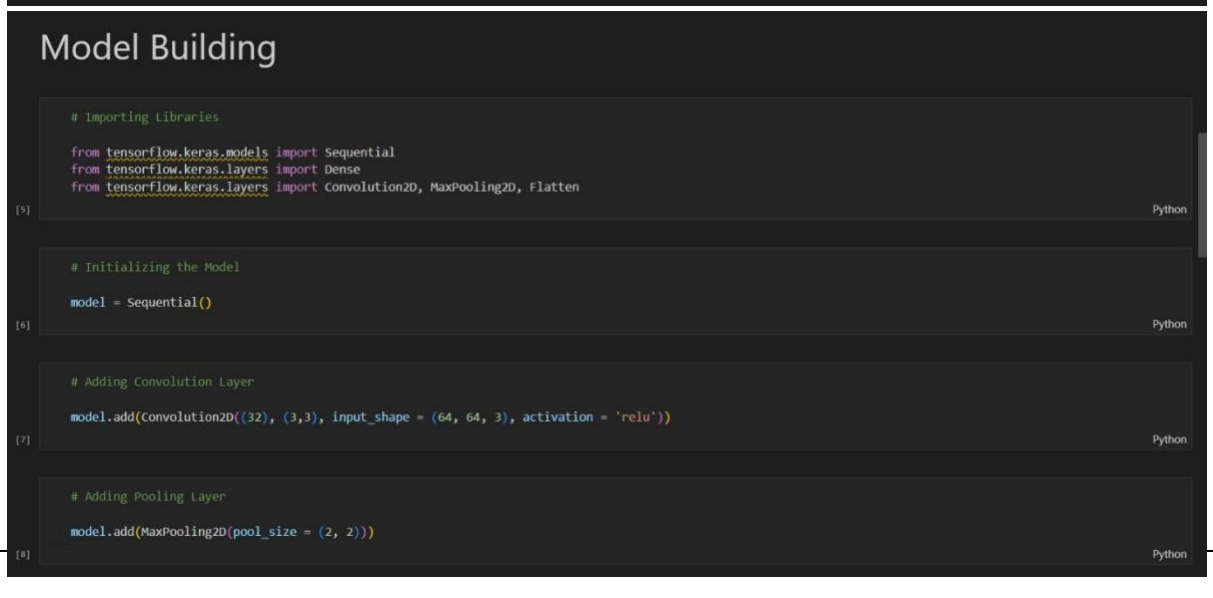
```
# Image Augmentation

train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```
[2] Python

```
# Loading train and test set

X_train = train_datagen.flow_from_directory(r"D:\Maheshfiles\Studies\Smart Bridge\AI-ML-DL Project\Dataset\training_set", target_size = (64, 64), batch_size =
X_test = test_datagen.flow_from_directory(r"D:\Maheshfiles\Studies\Smart Bridge\AI-ML-DL Project\Dataset\test_set", target_size = (64, 64), batch_size = 32, cl
```
[3] Python

```
Found 15750 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

```
# checking indices

X_train.class_indices
```
[4] Python

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



Model Building

```
# Importing Libraries

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten
```
[5] Python

```
# Initializing the Model

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

```
# Adding Convolution Layer

model.add(Convolution2D((32), (3,3), input_shape = (64, 64, 3), activation = 'relu'))
```
[7] Python

```
# Adding Pooling Layer

model.add(MaxPooling2D(pool_size = (2, 2)))
```
[8] Python

```python
# Adding Flatten Layer

model.add(Flatten())
```

[9] Python

```python
# Adding Hidden Layer

model.add(Dense(units = 512, kernel_initializer = 'random_uniform', activation = 'relu'))
```

[10] Python

```python
# Adding Output Layer

model.add(Dense(units = 9, kernel_initializer = 'random_uniform', activation = 'softmax'))
```

[11] Python

```python
# Compile the model

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

[12] Python

```python
# Fiiting the model

model.fit_generator(X_train, steps_per_epoch = 24, epochs = 10, validation_data = X_test, validation_steps = 40)
```

] Python

```
C:\Users\mahes\AppData\Local\Temp/ipykernel_10216/1270027362.py:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
  model.fit_generator(X_train, steps_per_epoch = 24, epochs = 10, validation_data = X_test, validation_steps = 40)

Epoch 1/10
24/24 [==============================] - 26s 1s/step - loss: 1.4863 - accuracy: 0.5625 - val_loss: 0.6678 - val_accuracy: 0.7930
Epoch 2/10
24/24 [==============================] - 21s 878ms/step - loss: 0.5226 - accuracy: 0.8385 - val_loss: 0.3198 - val_accuracy: 0.9336
Epoch 3/10
24/24 [==============================] - 18s 759ms/step - loss: 0.3561 - accuracy: 0.8854 - val_loss: 0.3711 - val_accuracy: 0.9328
Epoch 4/10
24/24 [==============================] - 17s 711ms/step - loss: 0.2102 - accuracy: 0.9362 - val_loss: 0.2478 - val_accuracy: 0.9492
Epoch 5/10
24/24 [==============================] - 15s 638ms/step - loss: 0.1726 - accuracy: 0.9570 - val_loss: 0.2474 - val_accuracy: 0.9469
Epoch 6/10
24/24 [==============================] - 16s 648ms/step - loss: 0.1651 - accuracy: 0.9505 - val_loss: 0.2897 - val_accuracy: 0.9617
Epoch 7/10
24/24 [==============================] - 13s 560ms/step - loss: 0.1277 - accuracy: 0.9609 - val_loss: 0.2441 - val_accuracy: 0.9586
Epoch 8/10
24/24 [==============================] - 13s 543ms/step - loss: 0.0985 - accuracy: 0.9714 - val_loss: 0.2331 - val_accuracy: 0.9539
Epoch 9/10
24/24 [==============================] - 13s 528ms/step - loss: 0.0995 - accuracy: 0.9701 - val_loss: 0.2301 - val_accuracy: 0.9609
Epoch 10/10
24/24 [==============================] - 12s 503ms/step - loss: 0.0913 - accuracy: 0.9779 - val_loss: 0.2053 - val_accuracy: 0.9742

<keras.callbacks.History at 0x1d9801fe9d0>
```

```python
# Saving the model

model.save('aslpng1.h5')
```

[14] Python

TESTING THE DATASET

```
# Loading Model

model = load_model("aslpng1.h5")
```

[7] Python

```
video = cv2.VideoCapture(0)
index = ['A','B','C','D','E','F','G','H','I']
```

[8] Python

```
while True:
    success, frame = video.read()
    cv2.imwrite('frame.jpg', frame)
    img = image.load_img('frame.jpg', target_size = (64, 64))

    x = image.img_to_array(img)
    x = cv2.cvtColor(x, cv2.COLOR_BGR2HSV)
    a = x.array_to_img(x)
    cv2.imshow("")
    x = np.expand_dims(x, axis = 0)
    pred = np.argmax(model.predict(x), axis = 1)

    y = pred[0]

    copy = frame.copy()


    cv2.rectangle(copy, (320, 100), (620, 400), (255, 0, 0), 5)
    cv2.putText(frame, "The Predicted Alphabet : " + str(index[y]), (100, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)
    cv2.imshow('frame', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video.release()
cv2.destroyAllWindows()
```

[10] Python

```
... Output exceeds the size limit. Open the full output data in a text editor
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 16ms/step
```

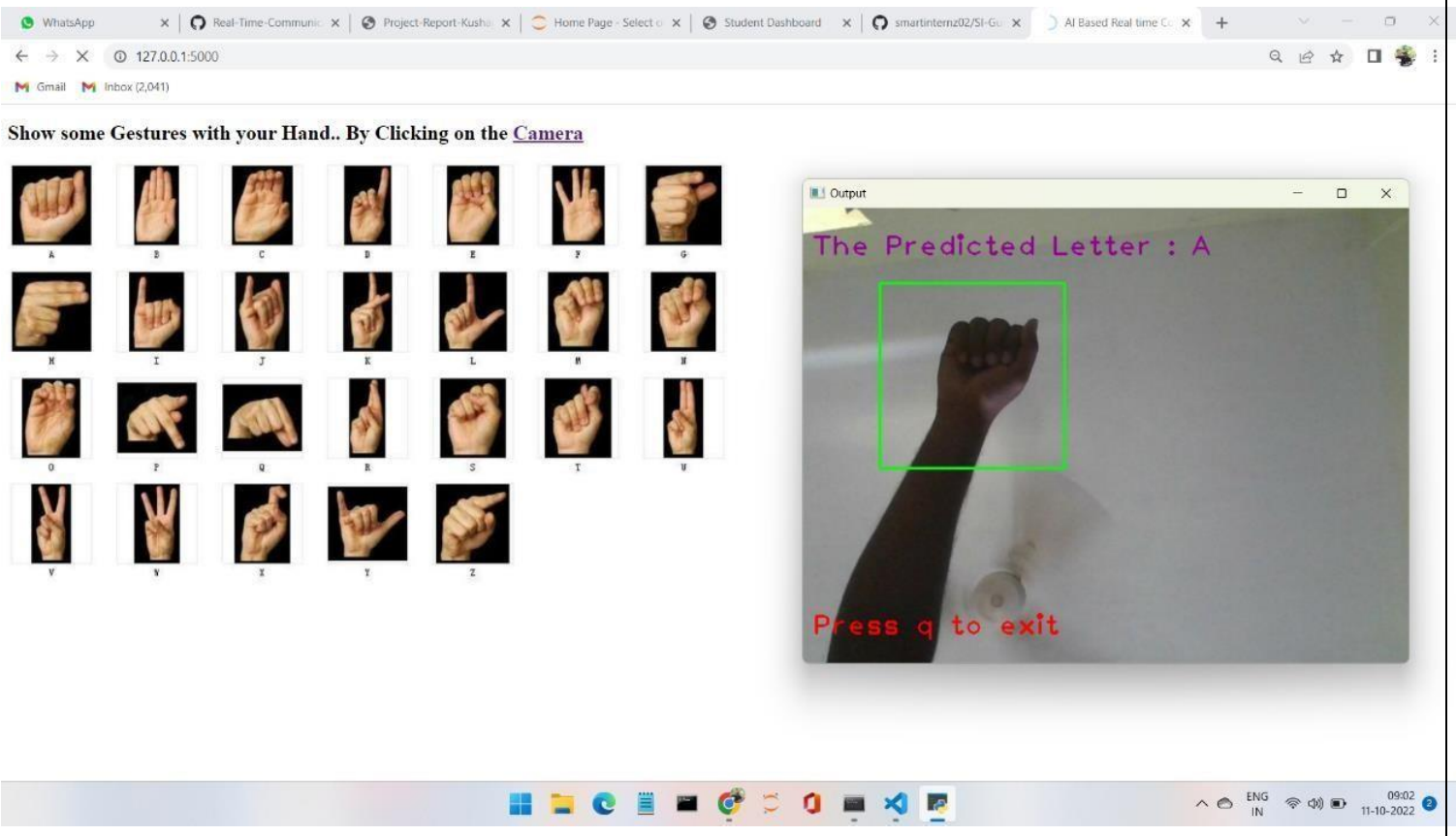
8.Testing:

8.1 test cases:

Test case	Expected output	Actual output	Test results and comments
Registration	Sign page access	Sign page access	Positive User name:Priya age 25 complaint: Deaf mute
Login	Login access	Login access	Positive User name:kaviya age 18 no complaint
Dashboard	Home page access	Home page access	Positive User name:sri devi age 20 no complaint
Speech to text	Speech gets converted to text	Speech gets converted to text	Positive User name:hema age 25 normal
Sign language to text	Sign language gets converted to text	Sign languagegets converted to text	Positive User name:sri devi age 20 no complaint

9.Results:

The proposed procedure was implemented and tested with set of images. The set of 15750 images of Alphabets from “A” to “I” are used for training database and a set of 2250 images of Alphabets from “A” to “I” are used for testing database. Once the gesture is recognize the equivalent Alphabet is shown on the screen.



10. Advantages and disadvantages:

Advantages:

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

Disadvantages:

- The current model only works from alphabets A to I.
- In absence of gesture recognition, alphabets from J cannot be identified as they requiresome kind of gesture input from the user.
- As the quantity/quality of images in the dataset is low, the accuracy is not great, but that can easily be improved by change in dataset.

11. APPLICATION:

- It will contribute to the development of improved communication for the deafened. Themajority of people are unable to communicate via sign language, which creates a barrierto communication.
- As a result, others will be able to learn and comprehend sign language and communicate with the deaf and dumb via the web app.
- According to scientific research, learning sign language improves cognitive abilities, attention span, and creativity.

12. CONCLUSION:

Sign language is a useful tool for facilitating communication between deaf and hearing people. Because it allows for two-way communication, the system aims to bridge the communication gap between deaf people and rest of society. The proposed methodology translates language into English alphabets that are understandable to humans. This system sends hand gestures to the model, who recognize them and displays the equivalent Alphabet on the screen. Deaf-mute people can use their hands to perform signlanguage, which will then be converted into alphabets.

12. FUTURE SCOPE:

Having a technology that can translate hand sign language to its corresponding

alphabet is a game changer in the field of communication and AI for the specially abled people such as deaf and dumb. With introduction of gesture recognition, the web app can easily be expanded to recognize letters beyond ‘I’, digits and other symbols plus gesture recognition can also allow controlling of software/hardware interfaces**.**

13. APPENDIX

```python
# Importing Libraries

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

[1] Python

```python
# Image Augmentation

train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

[2] Python

```python
# Loading train and test set

X_train = train_datagen.flow_from_directory(r"D:\Maheshfiles\Studies\Smart Bridge\AI-ML-DL Project\Dataset\training_set", target_size = (64, 64), batch_size =
X_test = test_datagen.flow_from_directory(r"D:\Maheshfiles\Studies\Smart Bridge\AI-ML-DL Project\Dataset\test_set", target_size = (64, 64), batch_size = 32, cl
```

[3] Python

```
Found 15750 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

```python
# checking indices

X_train.class_indices
```

[4] Python

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

Model Building

```python
# Importing Libraries

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten
```

[5] Python

```python
# Initializing the Model

model = Sequential()
```

[6] Python

```python
# Adding Convolution Layer

model.add(Convolution2D((32), (3,3), input_shape = (64, 64, 3), activation = 'relu'))
```

[7] Python

```python
# Adding Pooling Layer

model.add(MaxPooling2D(pool_size = (2, 2)))
```

[8] Python

```python
# Adding Flatten Layer

model.add(Flatten())
```

[9] Python

```python
# Adding Hidden Layer

model.add(Dense(units = 512, kernel_initializer = 'random_uniform', activation = 'relu'))
```

[10] Python

```python
# Adding Output Layer

model.add(Dense(units = 9, kernel_initializer = 'random_uniform', activation = 'softmax'))
```

[11] Python

```python
# Compile the model

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

[12] Python

```python
# Fiiting the model

model.fit_generator(X_train, steps_per_epoch = 24, epochs = 10, validation_data = X_test, validation_steps = 40)
```

Python

```
C:\Users\mahes\AppData\Local\Temp/ipykernel_10216/1270027362.py:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
  model.fit_generator(X_train, steps_per_epoch = 24, epochs = 10, validation_data = X_test, validation_steps = 40)

Epoch 1/10
24/24 [==============================] - 26s 1s/step - loss: 1.4863 - accuracy: 0.5625 - val_loss: 0.6678 - val_accuracy: 0.7930
Epoch 2/10
24/24 [==============================] - 21s 878ms/step - loss: 0.5226 - accuracy: 0.8385 - val_loss: 0.3198 - val_accuracy: 0.9336
Epoch 3/10
24/24 [==============================] - 18s 759ms/step - loss: 0.3561 - accuracy: 0.8854 - val_loss: 0.3711 - val_accuracy: 0.9328
Epoch 4/10
24/24 [==============================] - 17s 711ms/step - loss: 0.2102 - accuracy: 0.9362 - val_loss: 0.2478 - val_accuracy: 0.9492
Epoch 5/10
24/24 [==============================] - 15s 638ms/step - loss: 0.1726 - accuracy: 0.9570 - val_loss: 0.2474 - val_accuracy: 0.9469
Epoch 6/10
24/24 [==============================] - 16s 648ms/step - loss: 0.1651 - accuracy: 0.9505 - val_loss: 0.2897 - val_accuracy: 0.9617
Epoch 7/10
24/24 [==============================] - 13s 560ms/step - loss: 0.1277 - accuracy: 0.9609 - val_loss: 0.2441 - val_accuracy: 0.9586
Epoch 8/10
24/24 [==============================] - 13s 543ms/step - loss: 0.0985 - accuracy: 0.9714 - val_loss: 0.2331 - val_accuracy: 0.9539
Epoch 9/10
24/24 [==============================] - 13s 528ms/step - loss: 0.0995 - accuracy: 0.9701 - val_loss: 0.2301 - val_accuracy: 0.9609
Epoch 10/10
24/24 [==============================] - 12s 503ms/step - loss: 0.0913 - accuracy: 0.9779 - val_loss: 0.2053 - val_accuracy: 0.9742

<keras.callbacks.History at 0x1d9801fe9d0>
```

```python
# Saving the model

model.save('aslpng1.h5')
```

[14]

Python

```python
# Importing Libarries

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

[1]

Python

```python
# loading model

model = load_model('aslpng1.h5')
```

[2]

Python

```python
from skimage.transform import resize
def detect(frame):
    img = resize(frame, (64, 64, 3))
    img = np.expand_dims(img, axis = 0)
    if np.max(img) > 1:
        img = img/255.0
    prediction = model.predict(img)
    print(prediction)
    return prediction
```

[3]

Python

```python
frame = cv2.imread(r"D:\Maheshfiles\Studies\Smart Bridge\AI-ML-DL Project\Dataset\training_set\D\16.png")
data = detect(frame)
```

[4] Python

```
1/1 [==============================] - 0s 266ms/step
[[3.9748478e-08 1.2755189e-05 1.0463478e-08 9.9853325e-01 2.6569789e-06
  1.3680419e-05 4.5120544e-08 1.8048374e-07 1.4373119e-03]]
```

```python
index = ['A','B','C','D','E','F','G','H','I']
index[np.argmax(data)]
```

[5] Python

```
'D'
```

OpenCV

```python
# Importing Libraries

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

[6] Python

```python
# Loading Model

model = load_model("aslpng1.h5")
```

[7] Python

```python
video = cv2.VideoCapture(0)
index = ['A','B','C','D','E','F','G','H','I']
```

[8] Python

```python
while True:
    success, frame = video.read()
    cv2.imwrite('frame.jpg', frame)
    img = image.load_img('frame.jpg', target_size = (64, 64))

    x = image.img_to_array(img)
    x = cv2.cvtColor(x, cv2.COLOR_BGR2HSV)
    a = x.array_to_img(x)
    cv2.imshow("")
    x = np.expand_dims(x, axis = 0)
    pred = np.argmax(model.predict(x), axis = 1)

    y = pred[0]

    copy = frame.copy()


    cv2.rectangle(copy, (320, 100), (620, 400), (255, 0, 0), 5)
    cv2.putText(frame, "The Predicted Alphabet : " + str(index[y]), (100, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)
    cv2.imshow('frame', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video.release()
cv2.destroyAllWindows()
```

[10] Python

```
Output exceeds the size limit. Open the full output data in a text editor
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 16ms/step
```

```
# Saving the model

model.save('aslpng1.h5')
```
[39] Python

```
!tar -zcvf ai-based-real-time-classification-model.tgz aslpng1.h5
```
[52] Python

```
aslpng1.h5
```

```
!pip install watson-machine-learning-client
```
[41] Python

```
Collecting watson-machine-learning-client
  Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)
     |████████████████████████████████| 538 kB 23.2 MB/s eta 0:00:01
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.6.15)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client)
(1.21.41)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client)
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url" : "https://us-south.ml.cloud.ibm.com",
    "apikey" : "T7rpH1KfTn-s-zfDCmTyeArxYrcvFGHFV21qTDW5pf5x"
}
client = APIClient(wml_credentials)
```
[42] Python

```
def guid_space_name(client, ai_based_real_time_communication_deploy_space):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if item['entity']['name'] == ai_based_real_time_communication_deploy_space)['metadata']['id'])
```
[43] Python

```
client.spaces.get_details()
```
[45] Python

```
Output exceeds the size limit. Open the full output data in a text editor
{'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:us-south:a/5be23fa7fba94c8aa2e3db0b2a4db8d2:04f159f4-9ffb-4e0d-b70b-2a1f3b216970::',
     'guid': '04f159f4-9ffb-4e0d-b70b-2a1f3b216970',
     'name': 'Watson Machine Learning-av',
     'type': 'machine_learning'}],
   'description': '',
   'name': 'ai_based_real_time_communication_deploy_space',
   'scope': {'bss_account_id': '5be23fa7fba94c8aa2e3db0b2a4db8d2'},
   'stage': {'production': False},
   'status': {'state': 'active'},
   'storage': {'properties': {'bucket_name': '0a42d73b-35af-4f9d-92da-4a84147fcb1c',
     'bucket_region': 'us-south',
```

```
space_id = guid_space_name(client, 'ai_based_real_time_communication_deploy_space')
space_id
```

[47] Python

```
'1853d74e-ca3c-4075-81e3-d5cdd0741a52'
```

```
client.set.default_space(space_id)
```

[48] Python

```
'SUCCESS'
```

```
client.software_specifications.list(100)
```

[50] Python

Output exceeds the size limit. Open the full output data in a text editor

NAME	ASSET_ID	TYPE
default_py3.6	0062b8c9-8b7d-44a0-a9b9-46c416adcbd9	base
kernel-spark3.2-scala2.12	020d69ce-7ac1-5e68-ac1a-31189867356a	base
pytorch-onnx_1.3-py3.7-edt	069ea134-3346-5748-b513-49120e15d288	base
scikit-learn_0.20-py3.6	09c5a1d0-9c1e-4473-a344-eb7b665ff687	base
spark-mllib_3.0-scala_2.12	09f4cff0-90a7-5899-b9ed-1ef348aebdee	base
pytorch-onnx_rt22.1-py3.9	0b848dd4-e681-5599-be41-b5f6fccc6471	base
ai-function_0.1-py3.6	0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda	base
shiny-r3.6	0e6e79df-875e-4f24-8ae9-62dcc2148306	base
tensorflow_2.4-py3.7-horovod	1092590a-307d-563d-9b62-4eb7d64b3f22	base
pytorch_1.1-py3.6	10ac12d6-6b30-4ccd-8392-3e922c096a92	base
tensorflow_1.15-py3.6-ddl	111e41b3-de2d-5422-a4d6-bf776828c4b7	base

```
software_space_id = client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_id
```

[51] Python

```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

```
model_details = client.repository.store_model(model = 'ai-based-real-time-classification-model.tgz', meta_props = {
    client.repository.ModelMetaNames.NAME:"CNN Model Buiding",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_id
})
```

[53] Python

```
model_id = client.repository.get_model_id(model_details)
model_id
```

[54] Python

```
'59b18265-3a03-47d3-b2d8-d9a0c5106f05'
```

```
client.repository.download(model_id, 'ai-based-real-time-classification-model.h5')
```

[56] Python

```
Successfully saved model content to file: 'ai-based-real-time-classification-model.h5'

'/home/wsuser/work/ai-based-real-time-classification-model.h5'
```

```
pip install ibm_watson_machine_learning
```

[1] Python

```
Requirement already satisfied: ibm_watson_machine_learning in c:\users\mahes\anaconda3\lib\site-packages (1.0.253)
Requirement already satisfied: urllib3 in c:\users\mahes\anaconda3\lib\site-packages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: requests in c:\users\mahes\anaconda3\lib\site-packages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: certifi in c:\users\mahes\anaconda3\lib\site-packages (from ibm_watson_machine_learning) (2021.10.8)
Requirement already satisfied: ibm-cos-sdk==2.11.* in c:\users\mahes\anaconda3\lib\site-packages (from ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: tabulate in c:\users\mahes\anaconda3\lib\site-packages (from ibm_watson_machine_learning) (0.9.0)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in c:\users\mahes\anaconda3\lib\site-packages (from ibm_watson_machine_learning) (1.3.4)
Requirement already satisfied: packaging in c:\users\mahes\anaconda3\lib\site-packages (from ibm_watson_machine_learning) (21.0)
Requirement already satisfied: importlib-metadata in c:\users\mahes\anaconda3\lib\site-packages (from ibm_watson_machine_learning) (4.8.1)
Requirement already satisfied: lomond in c:\users\mahes\anaconda3\lib\site-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in c:\users\mahes\anaconda3\lib\site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning)
(0.10.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in c:\users\mahes\anaconda3\lib\site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning)
(2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in c:\users\mahes\anaconda3\lib\site-packages (from ibm-cos-sdk==2.11.*-
>ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\mahes\anaconda3\lib\site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*-
>ibm_watson_machine_learning) (2.8.2)
Requirement already satisfied: numpy>=1.17.3 in c:\users\mahes\anaconda3\lib\site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.20.3)
Requirement already satisfied: pytz>=2017.3 in c:\users\mahes\anaconda3\lib\site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2021.3)
Requirement already satisfied: idna<4,>=2.5 in c:\users\mahes\anaconda3\lib\site-packages (from requests->ibm_watson_machine_learning) (3.2)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\mahes\anaconda3\lib\site-packages (from requests->ibm_watson_machine_learning) (2.0.4)
Requirement already satisfied: zipp>=0.5 in c:\users\mahes\anaconda3\lib\site-packages (from importlib-metadata->ibm_watson_machine_learning) (3.6.0)
Requirement already satisfied: six>=1.10.0 in c:\users\mahes\anaconda3\lib\site-packages (from lomond->ibm_watson_machine_learning) (1.16.0)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\mahes\anaconda3\lib\site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url" : "https://us-south.ml.cloud.ibm.com",
    "apikey" : "T7rpH1KfTn-s-zfDCmTyeArxYrcvFGHFV21qTDW5pf5x"
}
client = APIClient(wml_credentials)
```

[2] Python

```
def guid_space_name(client, ai_based_real_time_communication_deploy_space):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if item['entity']['name'] == ai_based_real_time_communication_deploy_space)['metadata']['id'])
```

[3] Python

```
space_id = guid_space_name(client, 'ai_based_real_time_communication_deploy_space')
space_id
```

[4] Python

```
'1853d74e-ca3c-4075-81e3-d5cdd0741a52'
```

```
client.set.default_space(space_id)
```

[5] Python

```
'SUCCESS'
```

```
client.repository.download('59b18265-3a03-47d3-b2d8-d9a0c5106f05', 'ai-based-real-time-classification-model.h5')
```

[6] Python

```
Successfully saved model content to file: 'ai-based-real-time-classification-model.h5'

'D:\\Maheshfiles\\Studies\\Smart Bridge\\AI-ML-DL Project/ai-based-real-time-classification-model.h5'
```

webstreaming.py

```python
from flask import Flask,render_template,request
import cv2
from keras.models import load_model
import numpy as np
from gtts import gTTS
import os
from keras.preprocessing import image
from skimage.transform import resize
from playsound import playsound
app = Flask(__name__)

model=load_model("aslpng1.h5")

vals = ['A', 'B','C','D','E','F','G','H','I']

@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')
@app.route('/index', methods=['GET'])
def home():
    return render_template('index.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
        print("[INFO] starting video stream...")
        vs = cv2.VideoCapture(0)

        (W, H) = (None, None)

        while True:
            (grabbed, frame) = vs.read()

            if not grabbed:
                break

            if W is None or H is None:
                (H, W) = frame.shape[:2]
            output = frame.copy()
            # r = cv2.selectROI("Slect", output)
            # print(r)
            cv2.rectangle(output, (81, 79), (276,274), (0,255,0), 2)
            frame = frame[81:276, 79:274]
            frame = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
            _, frame = cv2.threshold(frame, 95, 255, cv2.THRESH_BINARY_INV)
            frame = cv2.cvtColor(frame, cv2.COLOR_GRAY2RGB)


            img = resize(frame,(64,64,3))
            img = np.expand_dims(img,axis=0)
            if(np.max(img)>1):
                img = img/255.0


            result = np.argmax(model.predict(img))
            index=['A', 'B','C','D','E','F','G','H','I']
            result=str(index[result])


            cv2.putText(output, "The Predicted Letter : {}".format(result), (10, 50), cv2.FONT_HERSHEY_PLAIN,
                        2, (150,0,150), 2)
            cv2.putText(output, "Press q to exit", (10,450), cv2.FONT_HERSHEY_PLAIN, 2, (0,0,255), 2)


            speech = gTTS(text = result, lang = 'en', slow = False)

            cv2.imshow("Output", output)
            key = cv2.waitKey(1) & 0xFF

            if key == ord("q"):
                break
```

```
        print("[INFO] cleaning up...")
        vs.release()
        cv2.destroyAllWindows()
        return render_template("index.html")


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

index.html ● webstreaming.py ●

templates > index.html > html

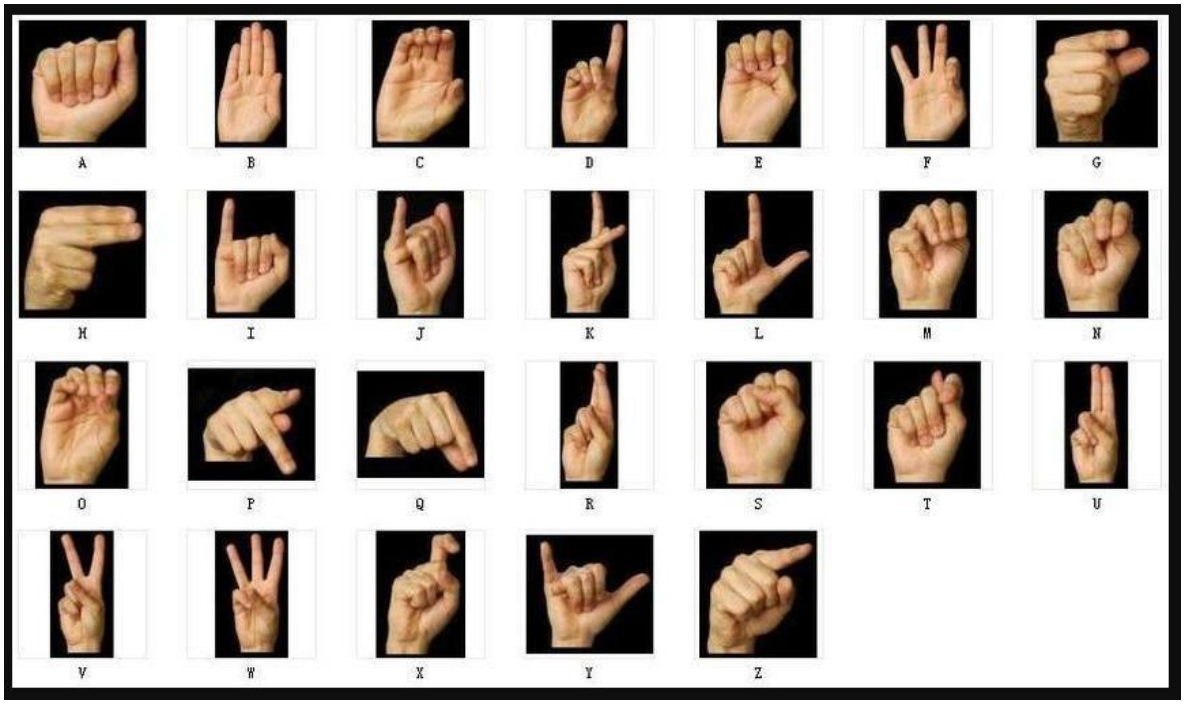
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>AI Based Real time Communication</title>
</head>
<body>
    <h2>Show some Gestures with your Hand.. By Clicking on the <a href="{{ url_for('predict') }}">Camera</a></h2>
    <div>
        <img src = 'https://www.researchgate.net/publication/274480405/figure/fig4/
        AS:668304138063878@1536347531535/26-sample-hand-gesture-images-for-26-letters-in-American-Sign-Language.jpg'  width = 50%>

    </div>
</body>
</html>
```

```
        if W is None or H is None:
            (H, W) = frame.shape[:2]
        output = frame.copy()
        # r = cv2.selectROI("Slect", output)
        # print(r)
        cv2.rectangle(output, (81, 79), (276,274), (0,255,0), 2)
        frame = frame[81:276, 79:274]
        frame = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
        _, frame = cv2.threshold(frame, 95, 255, cv2.THRESH_BINARY_INV)
        frame = cv2.cvtColor(frame, cv2.COLOR_GRAY2RGB)


        img = resize(frame,(64,64,3))
        img = np.expand_dims(img,axis=0)
        if(np.max(img)>1):
            img = img/255.0


        result = np.argmax(model.predict(img))
        index=['A', 'B','C','D','E','F','G','H','I']
        result=str(index[result])


        cv2.putText(output, "The Predicted Letter : {}".format(result), (10, 50), cv2.FONT_HERSHEY_PLAIN,
                    2, (150,0,150), 2)
        cv2.putText(output, "Press q to exit", (10,450), cv2.FONT_HERSHEY_PLAIN, 2, (0,0,255), 2)


        speech = gTTS(text = result, lang = 'en', slow = False)

        cv2.imshow("Output", output)
        key = cv2.waitKey(1) & 0xFF

        if key == ord("q"):
            break
```

webstreaming.py X

webstreaming.py > index

```
from flask import Flask,render_template,request
import cv2
from keras.models import load_model
import numpy as np
from gtts import gTTS
import os
from keras.preprocessing import image
from skimage.transform import resize
from playsound import playsound
app = Flask(__name__)

model=load_model("aslpng1.h5")

vals = ['A', 'B','C','D','E','F','G','H','I']

@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')
@app.route('/index', methods=['GET'])
def home():
    return render_template('index.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
        print("[INFO] starting video stream...")
        vs = cv2.VideoCapture(0)

        (W, H) = (None, None)

        while True:
            (grabbed, frame) = vs.read()

            if not grabbed:
                break
```


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

Downloading model from IBM Cloud:

```
72
73          print("[INFO] cleaning up...")
74          vs.release()
75          cv2.destroyAllWindows()
76          return render_template("index.html")
77
78
79  if __name__ == '__main__':
80          app.run(debug=False)
```

index.html ● webstreaming.py ●

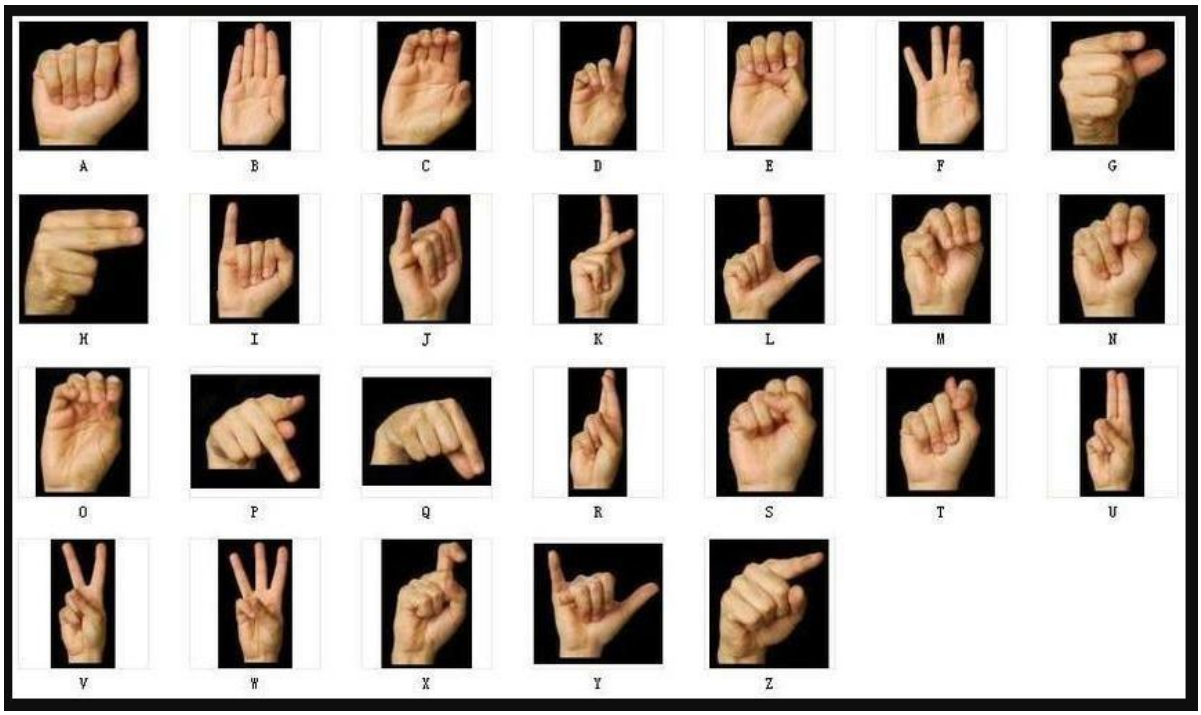
templates ❭ index.html ❭ html

```
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <title>AI Based Real time Communication</title>
6   </head>
7   <body>
8       <h2>Show some Gestures with your Hand.. By Clicking on the <a href="{{ url_for('predict') }}">Camera</a></h2>
9       <div>
10          <img src = 'https://www.researchgate.net/publication/274480405/figure/fig4/
11          AS:668304138063878@1536347531535/26-sample-hand-gesture-images-for-26-letters-in-American-Sign-Language.jpg'  width = 50%>
12
13      </div>
14  </body>
15  </html>
```

webstreaming.py

```python
from flask import Flask,render_template,request
import cv2
from keras.models import load_model
import numpy as np
from gtts import gTTS
import os
from keras.preprocessing import image
from skimage.transform import resize
from playsound import playsound
app = Flask(__name__)

model=load_model("aslpng1.h5")

vals = ['A', 'B','C','D','E','F','G','H','I']

@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')
@app.route('/index', methods=['GET'])
def home():
    return render_template('index.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
        print("[INFO] starting video stream...")
        vs = cv2.VideoCapture(0)

        (W, H) = (None, None)

        while True:
            (grabbed, frame) = vs.read()

            if not grabbed:
                break
```

```python
            if W is None or H is None:
                (H, W) = frame.shape[:2]
            output = frame.copy()
            # r = cv2.selectROI("Slect", output)
            # print(r)
            cv2.rectangle(output, (81, 79), (276,274), (0,255,0), 2)
            frame = frame[81:276, 79:274]
            frame = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
            _, frame = cv2.threshold(frame, 95, 255, cv2.THRESH_BINARY_INV)
            frame = cv2.cvtColor(frame, cv2.COLOR_GRAY2RGB)


            img = resize(frame,(64,64,3))
            img = np.expand_dims(img,axis=0)
            if(np.max(img)>1):
                img = img/255.0


            result = np.argmax(model.predict(img))
            index=['A', 'B','C','D','E','F','G','H','I']
            result=str(index[result])


            cv2.putText(output, "The Predicted Letter : {}".format(result), (10, 50), cv2.FONT_HERSHEY_PLAIN,
                        2, (150,0,150), 2)
            cv2.putText(output, "Press q to exit", (10,450), cv2.FONT_HERSHEY_PLAIN, 2, (0,0,255), 2)


            speech = gTTS(text = result, lang = 'en', slow = False)

            cv2.imshow("Output", output)
            key = cv2.waitKey(1) & 0xFF

            if key == ord("q"):
                break
```



GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-20594-1659755933.

VIDEO LINK:

https://drive.google.com/file/d/1lRHzIi0a62mTHLwpaRVEdo4F7P1r9p5G/view?usp=drivesdk