# LOYOLA INSTITUTE OF TECHNOLOGY

# REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

## TEAM ID: PNT2022TMID25606

**MENTOR NAME: VARADHARAJAN M**

**INDUSTRY MENTOR: DIVYA**

**TEAM LEADER AISHWARYA R-210919106006**
**TEAM MEMBER: KANIMOZHI V-210919106037**
**TEAM MEMBER: JAYANTHINI S-210919106034**
**TEAM MEMBER: DHARANI M-210919106022**

# Project Report Format

**1. INTRODUCTION**

1.1 Project Overview

1.2 Purpose

**2. LITERATURE SURVEY**

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

**3. IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

**4. REQUIREMENT ANALYSIS**

4.1 Functional requirement

4.2 Non-Functional requirements

**5. PROJECT DESIGN**

5.1 Data Flow Diagram

5.2 Solution & Technical Architecture

5.3 User Stories

**6. PROJECT PLANNING & SCHEDULING**

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

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

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

**8. TESTING**

8.1 Test Cases 8.2 User Acceptance Testing

**9. RESULTS**

9.1 Performance Metrics

**10. ADVANTAGES & DISADVANTAGES**

**11. CONCLUSION**

**12. FUTURE SCOPE**

**13. APPENDIX**

Source Code

GitHub & Project Demo Link

# 1. INTRODUCTION

### Project Overview

Real-time communications (RTC) is any mode of telecommunications in which all users can exchange information instantly or with negligible latency or transmission delays. In RTC, there is always a direct path between the source and the destination. Although the link might contain several intermediate nodes, the data goes from source to destination without being stored in between them. In contrast, asynchronous or timeshifting communications, such as email and voicemail, always involve some form of data storage between the source and the destination. In these cases, there is an anticipated delay between the transmission and receipt of the information.

### Purpose

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. 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. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

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. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

## Problem Statement Definition

Communication is the only medium by which we can share our thoughts or convey the message but communications between deaf-mute and a normal person has always ben 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. In emergency times conveying their message is very difficult

**problem:**

Vedha has difficulty in hearing. He uses sign language to communicate with others. But he can't able to communicate with normal people who don't understand sign language.

**Solution:**

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 the system enhances the user friendly experience.

**problem**:

Itam is a dumb by birth. He uses sign language to communicate with others. But he can't able to communicate with normal people who don't understand sign language.

**solution:**

To create people app for understanding sign language and convert into Speech signal as output for normal .

## 2. LITERATURE SURVEY

### Existing Problem

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. Communication 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 onhand sign language. In emergency times conveying their message is very difficult.

The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

## References

1. Koufos, K., EL Halou, K, Dianati, M., Higgins, M., Elmirghani, J, Imran, M. A., &Tafazolli, R. (2021). Trends in Intelligent Communication Systems: Review of Standards, Major Research Projects, and Identification of Research Gaps. Journal of Sensor and Actuator Networks, 10(4), 60.

2. Panda, G., Upadhyay, A. K., & Khandelwal, K. (2019). Artificial intelligence: A strategic disruption in public relations. Journal of Creative Communications, 14(3), 196-213.

3. Xu, G., Mu, Y., & Liu, J. (2017). Inclusion of artificial intelligence in communication networks and services. ITU J. ICT Discov. Spec, 1, 1-6.

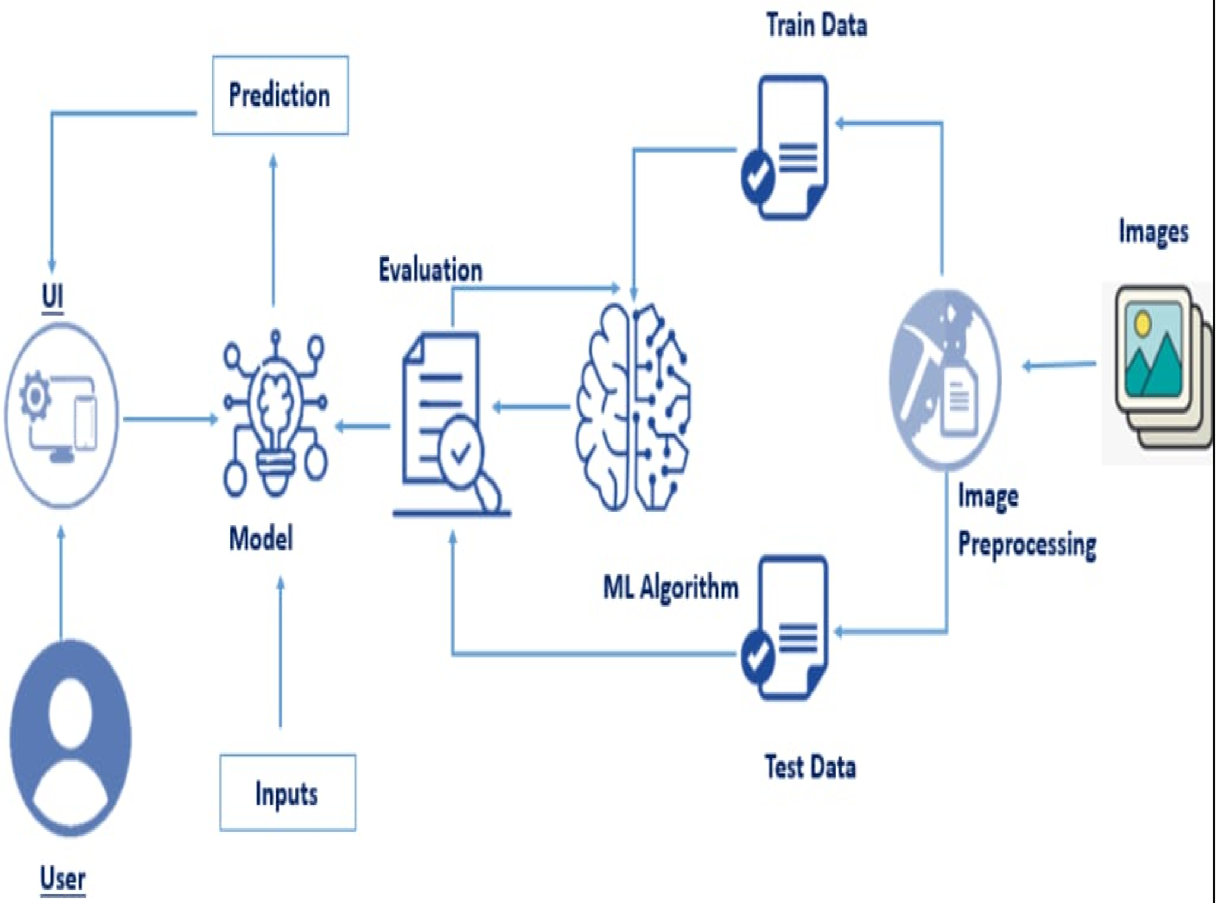
**Problem Statement Definition**

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. 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. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

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. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to humanunderstandable language and speech is given as output.

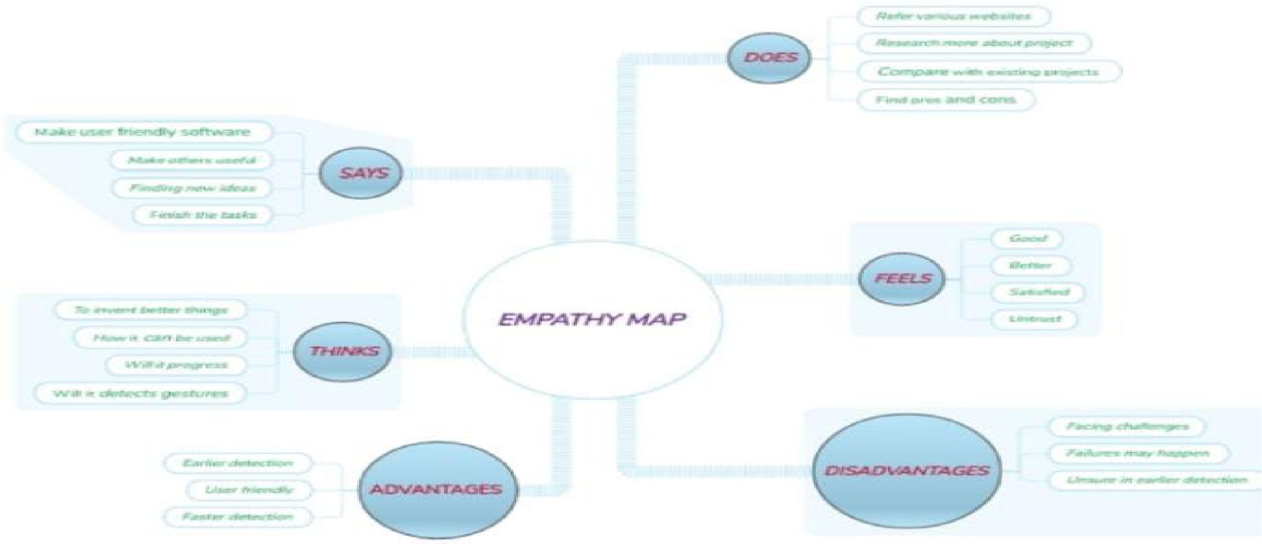
**PROPOSED SOLUTION**

| S.NO | PARAMETER | DESCRIPTION |
|---|---|---|
| 1 | PROBLEM STATEMENT | In our society, we have people with disabilities.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. In Emergency times conveying their message is very difficult |
| 2 | SOLUTION DESCRIPTION | The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. REAL TIME COMMUNICATION SYSTEM POWERED BY AL FOR SPECIALLY ABLED will be very useful to have a proper conversation between a normal person and an impaired person in any language |

| | | |
|---|---|---|
| 3 | NOVELTY/UNIQUENESS | This app converts the sign language into a human hearing voice in the cesired language to convey a message to normal people, as well as converts speech into understandable sign language for the deaf-mute |
| 4 | SOCIAL IMPACT | People with disabilities can drastically improve their everyday lives. |
| 5 | BUSINESS MODEL (REVENUE MODEL) |   Train Data<br>Prediction<br>Images<br>UI<br>Evaluation<br>Model<br>Image Preprocessing<br>ML Algorithm<br>Test Data<br>Inputs<br>User |

## EMPATHY MAP

Team ID: PNT2022TMID25606
Project name: REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED
Maximum mark: 4 MARKS



# REQUIREMENT ANALYSIS

## FUNCTIONAL REQUIREMENTS

| S.NO | FUNCTIONAL REQUIREMENTS | SUB REQUIRMENTS |
|---|---|---|
| FR-1 | User Registration | Registration through gmail or registration through moblie number |
| FR-2 | User confirmation | Confirmation via Email or Confirmation via OTP |
| FR-3 | System Reqirements | 1.moblie or PC or Laptop with webcam or camera<br>2.Minimum 1GB RAMand picture capability |
| FR-4 | Text conversion | Converts the sign language into a text using CNN model |
| FR-5 | sentence translation | To creat sentences by recognizing the signs and pauses in the video stream |
| FR-6 | Speech translation | TTS converts text into speech |

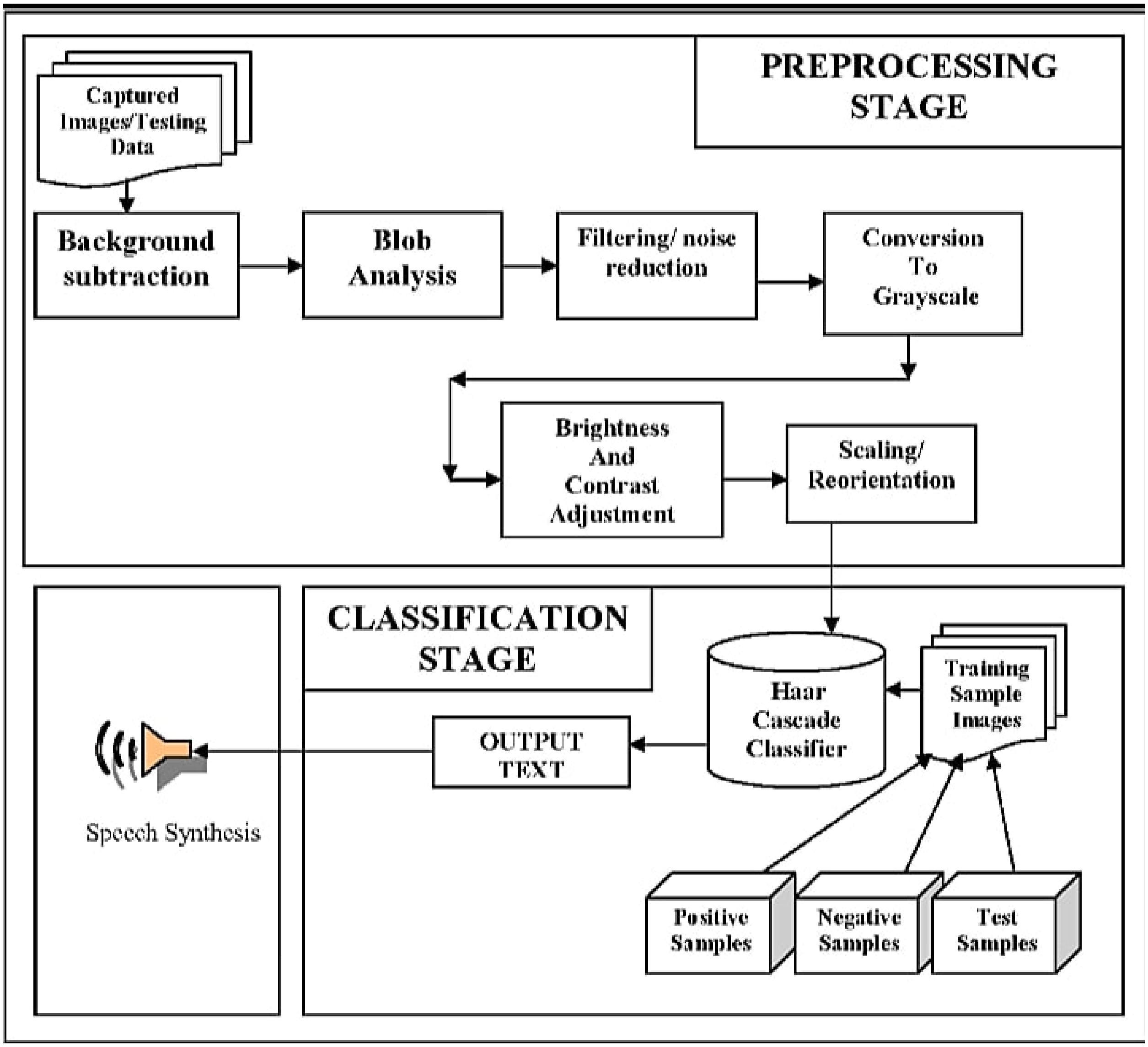
## NON FUNCTIONAL REQUIREMENTS

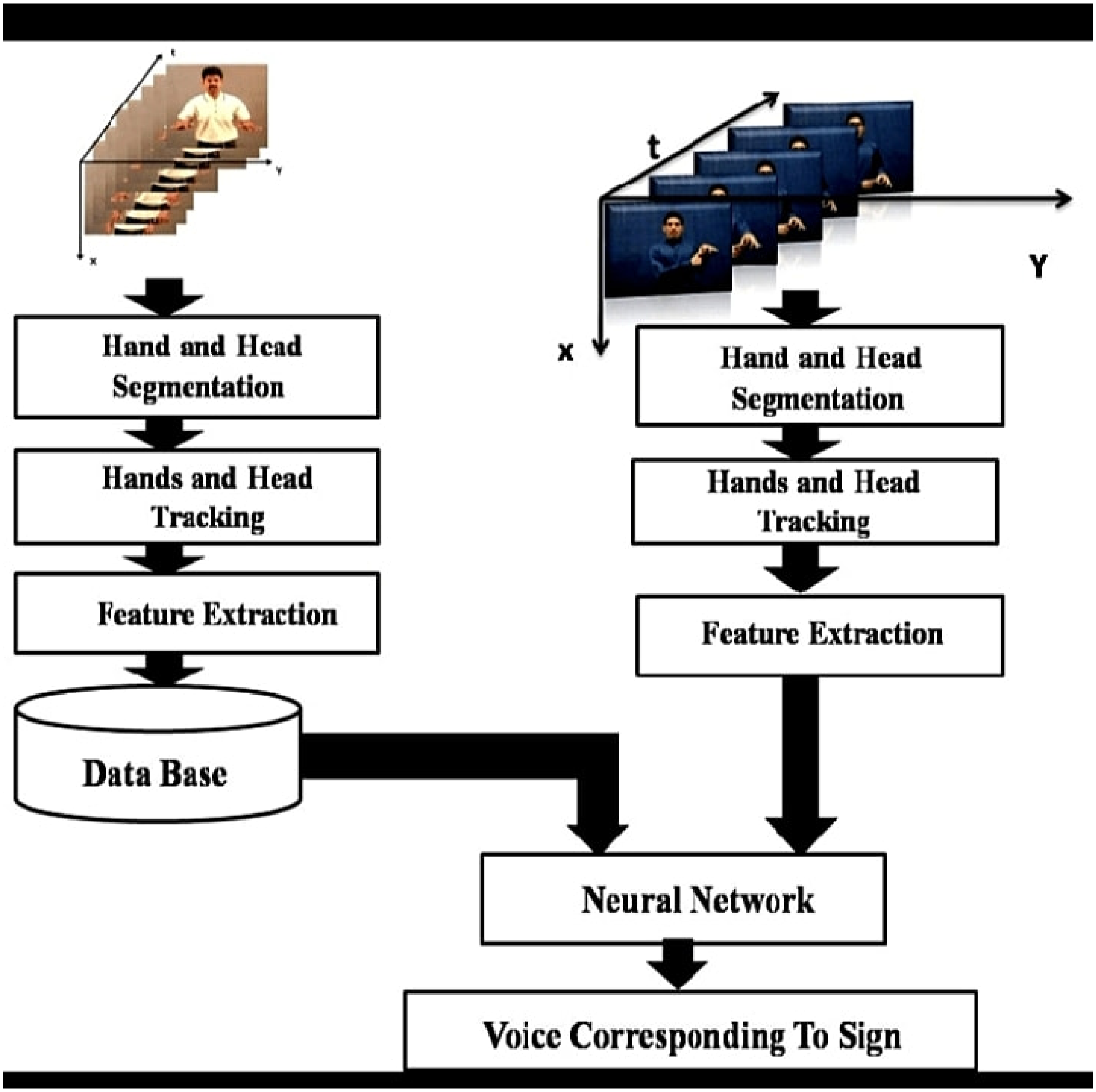
| **FR.NO** | **NON FUNCTIONAL REQUIREMENTS** | **DESCRIPTION** |
|---|---|---|
| NFR1 | Usability | Easy usable application for everyone.Especially useful for disabled person. It is user friendly |
| NFR-2 | Security | It is also a secured application and information and images are securely stored.It must be ensured that the privacy of user data be maintained and handled appropriately. |
| NFR-3 | Reliability | The translation of sign languages should be reliable. The accuracy of the system should be tested extensively to make sure that it is up to the mark. |
| NFR-4 | Performance | It's performance is consistency good .The processing should be done in considerable time so that the conversation can go on without waiting for the system's output. |

| | | |
|---|---|---|
| NFR-4 | Availability | It is afree accessible and Universal access.Since sign language is a almost same everywhere, the system can be used across the globe |

## project design:

## DATA FLOW DIAGRAM:




t
Y
x
t
Y
x
Hand and Head Segmentation
Hands and Head Tracking
Feature Extraction
Data Base
Hand and Head Segmentation
Hands and Head Tracking
Feature Extraction
Neural Network
Voice Corresponding To Sign

## Solution Architecture:

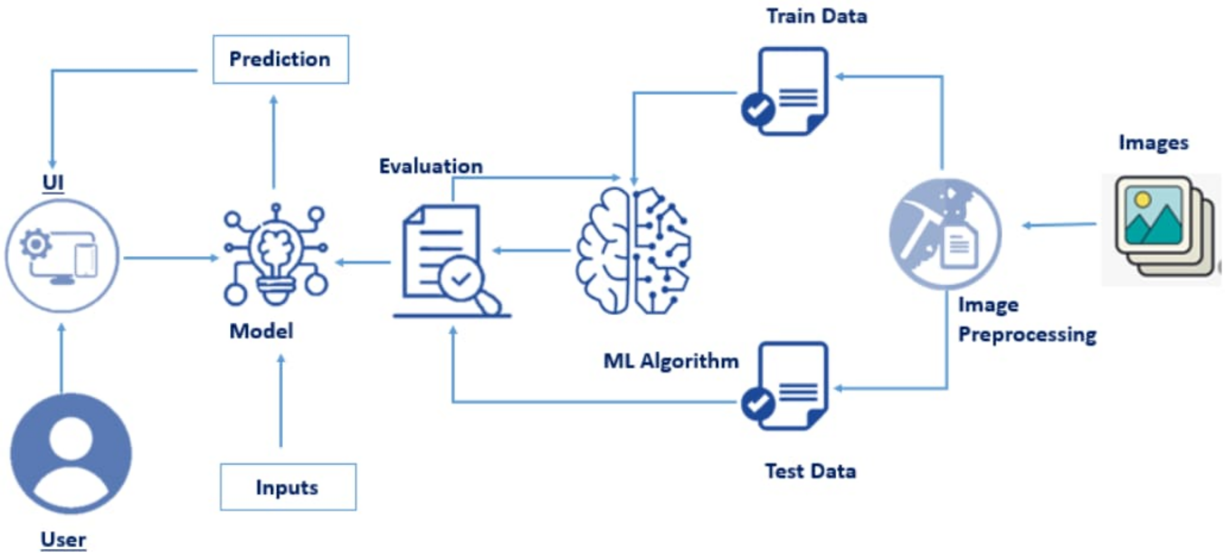
Solution architecture is a complex process - with many sub-processes - that bridges the gap between business problems and technology solutions. Its goals are to:

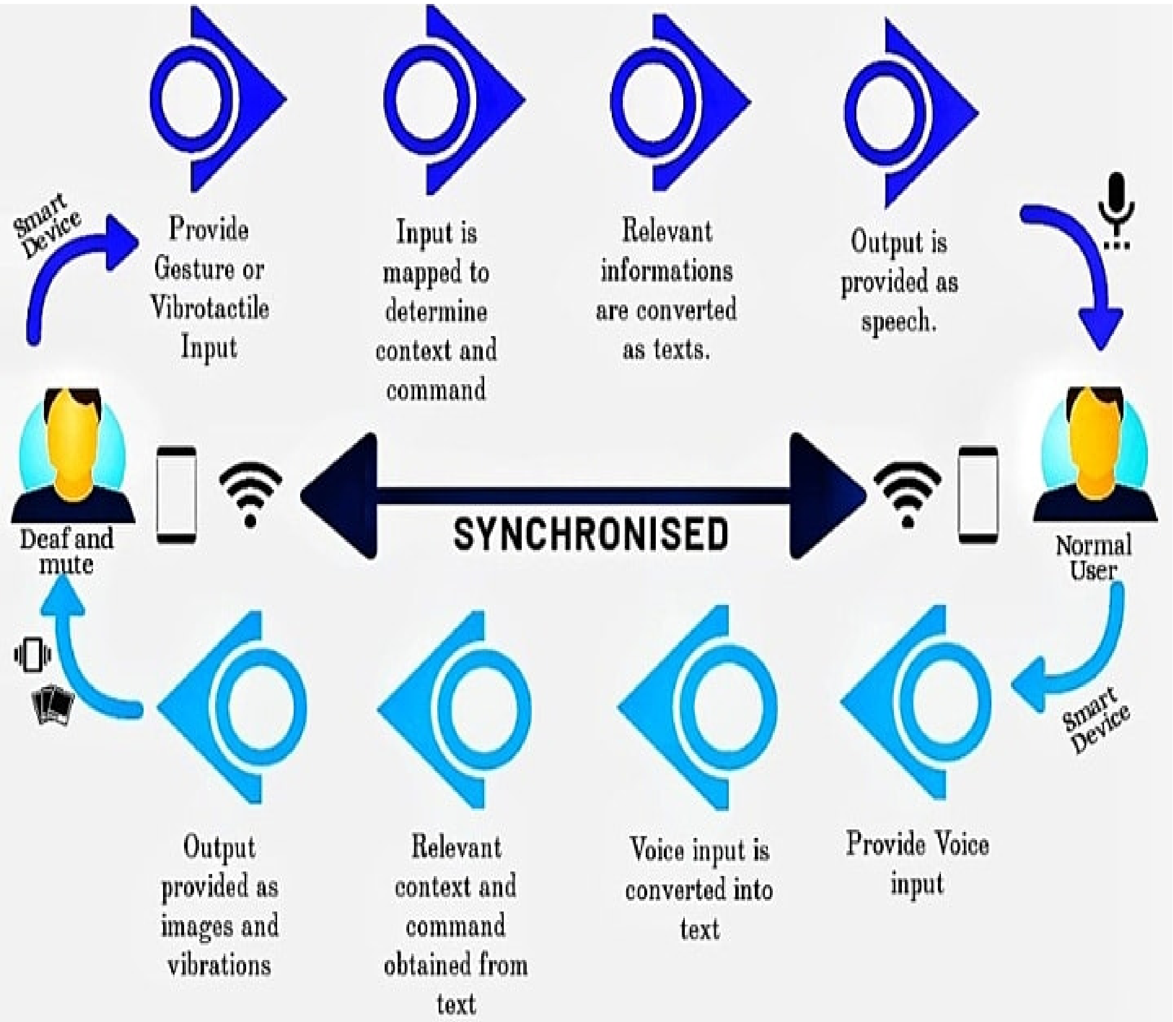
* Find the best tech solution to solve existing business problems.

* Describe the structure, characteristics, behavior, and other aspects of the  software to project stakeholders.

* Define features, development phases, and solution requirements.

*Provide specifications according to which the solution is defined, managed,and delivered.




Smart Device
Provide Gesture or Vibrotactile Input
Input is mapped to determine context and command
Relevant informations are converted as texts.
Output is provided as speech.
Deaf and mute
SYNCHRONISED
Normal User
Smart Device
Provide Voice input
Voice input is converted into text
Relevant context and command obtained from text
Output provided as images and vibrations

## MILESTONE ACTIVITY PLAN

| MILESTONE | FUNCTION | MILESTONE STORY NUMBER | STORY/TASK |
|---|---|---|---|
| MILESTONE-1 | Data collection | M1 | we're collecting dataset for building our project and creating two folders, une for training and another one for testing |
| MILESTONE-2 | Image preprocessing | M2 | Importing image data generator libraries and applying image data generator functionality totrain the test set |
| MILESTONE-3 | Model building | M3 | importing the model building libraries, Inibalizing 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 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 fiask application and the HIMLpages. |
| MILESTONE-6 | Train CNN model | M6 | Register for IBM Cloud and train ImageClassification Model |
| MILESTONE-7 | Final result | M7 | To ensure all the activities and resulting thefinal output. |

## SPRINT SCHEDULE:

| SPRINT | FUNCTIONAL REQUIREMENTS | USER STORY NUMBER | USER STORY/TASK | STORY POINTS | PRIORITY | TEAM MEMBERS |
|---|---|---|---|---|---|---|
| SPRINT-1 | Data collection | USN-1 | collect dataset | 9 | high | R.AISHWARYA |
| SPRINT-1 |  | USN-2 | image processing | 8 | medium | S.JAYANTHINI<br>V.KANIMOZHI |
| SPRINT-2 | Model building | USN-3 | import the required libraries, add the necessary layers and complete model | 10 | high | V.KANIMOZHI<br>M.DHARANI |
| SPRINT-2 |  | USN-4 | Training the image classification model using CNN | 7 | medium | M.DHARANI |
| SPRINT-3 | Training and testing | USN-5 | Training the model and testing the models performance | 9 | high | S.JAYANTHINI |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| SPRINT-4 | Implementation of the application | USN-6 | Converting the input sign language images into English alphabets | 8 | medium | R.AISHWARYA |

## PROJECT TRACKER:

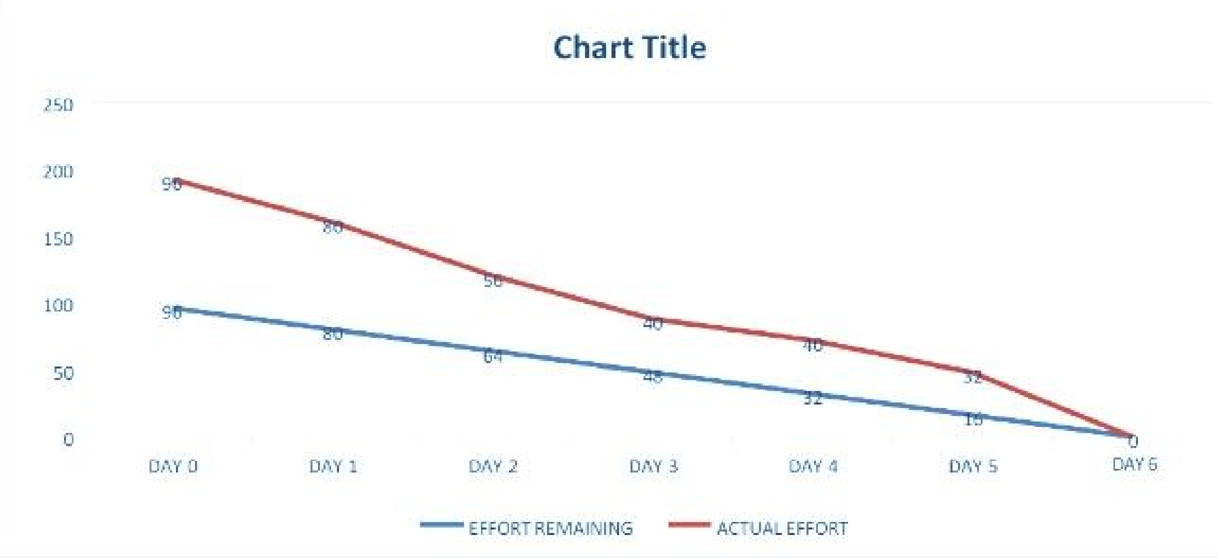
| SPRINT | TOTAL STORY POINTS | DURATION | SPRINT START DATE | SPRINT END DATE(PLANNED) | STORY POINTS COMPLETED | SPRINT RELEASE DATE(ACTUAL) |
|---|---|---|---|---|---|---|
| **SPRINT-1** | 10 | 6 days | 24 OCT 2022 | 29 OCT 2022 | **8** | 29 OCT 2022 |
| **SPRINT-2** | 10 | 6 days | 31 OCT 2022 | 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:

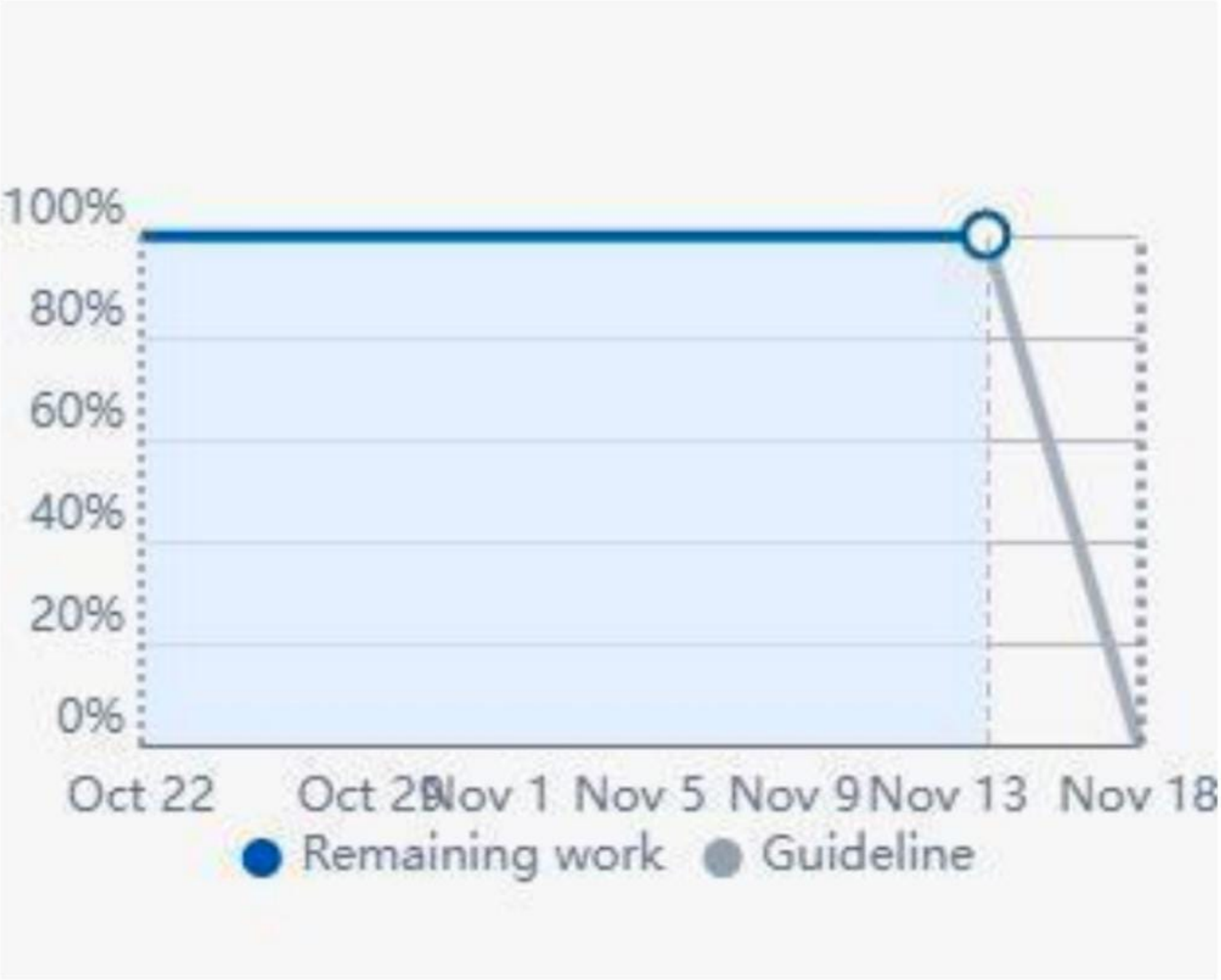
**AV=sprint duration/velocity**

**AV=6/10=0.6**

## BURNDOWN CHART

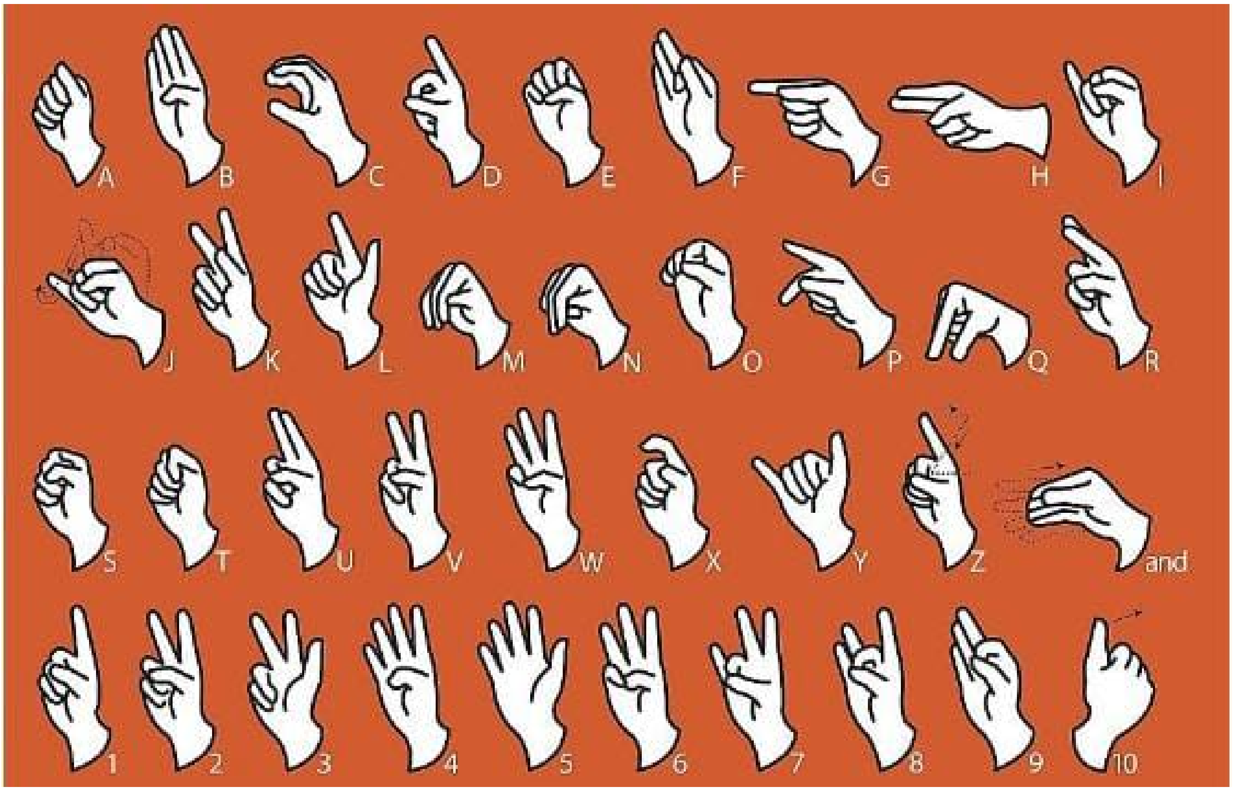


## SPRINT BURNDOWN CHART



## CODING AND SOLUTION

The user can choose which sign languageto read based on the different sign language standards that exist.



## MODEL BUILDING

```
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

#Creating the model
```

In [101]:

```
model=Sequential()
#Adding the layers
model.add(Convolution2D(32,(3,3),
input_shape=(64,64,1), activation = 'relu'))
model.add(MaxPooling2D(pool_size=(2,2))
model.add(Flatten())


#adding hidden layers
 model.add(Dense(400, activation='relu'))
 model.add(Dense(200, activation='relu'))
 model.add(Dense(100, activation='relu'))



            #Adding the output layer
model.add(Dense(9, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam',metrics=['accuracy'])
model.fit_generator(x_train, steps_per_epoch=30,
epochs=10,validation_data=x_test,validation_steps=50)
Epoch 1/10
/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 supports generators.
  """Entry point for launching an IPython kernel
30/30 [==============================] - ETA: 0s - loss: 0.0083-
accuracy:0.9957
WARNING:tensorflow:Your input ran out of data; interrupting
training.Make sure that yourdataset or generator can generate at least
`steps_per_epoch * epochs` batches (in this case, 50batches). You may
need touse the repeat() function when building your dataset.
30/30 [==============================] - 18s 587ms/step - loss: 0.0083 -
accuracy:0.9957 - val_loss: 0.2910 val_accuracy: 0.9693
Epoch 2/10
30/30 [==============================] - 12s 402ms/step - loss: 0.0081 -
accuracy:0.9980
Epoch 3/10
30/30 [==============================] - 12s 400ms/step - loss: 0.0102 -
accuracy:0.9963
Epoch 4/10
30/30 [==============================] - 12s 402ms/step - loss: 0.0049 -
accuracy:0.9993
Epoch 5/10
30/30 [==============================] - 12s 402ms/step - loss: 0.0030 -
accuracy:0.9997
Epoch 6/10
30/30 [==============================] - 12s 394ms/step - loss: 0.0019 -
accuracy:0.9997
Epoch 7/10
30/30 [==============================] - 12s 401ms/step - loss: 0.0081 -
accuracy:0.9973
```

```
Epoch 8/10
30/30 [=================================] - 12s 402ms/step - loss: 0.0124 -
accuracy:0.9960
Epoch 9/10
30/30 [=================================] - 12s 401ms/step - loss: 0.0070 -
accuracy:0.9987
Epoch 10/10
30/30 [=================================] - 12s 399ms/step - loss: 0.0089 -
accuracy:0.9973
```

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

TEST THE MODEL

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

```
model = load_model('/content/Real_time.h5')
```

```
img = image.load_img('/content/Dataset/test_set/H/107.png',target_size (100,100))img
```

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


img=image.load_img("/content/Dataset/test_set/H/107.png")
detect(img)
1/1 [==============================] - 0s 28ms/step
THE PREDICTED LETTER IS H

img = image.load_img('/content/Dataset/test_set/A/110.png')
pred=detect(img)
1/1 [==============================] - 0s 26ms/step
THE PREDICTED LETTER IS A

img=image.load_img('/content/Dataset/test_set/E/111.png')
detect(img)
1/1 [==============================] - 0s 30ms/step
THE PREDICTED LETTER IS E
```

## FEATURE 2

The communication gap between deaf and dumb people and the general public can be bridgedwith a mobile application.

**Mobile App:**

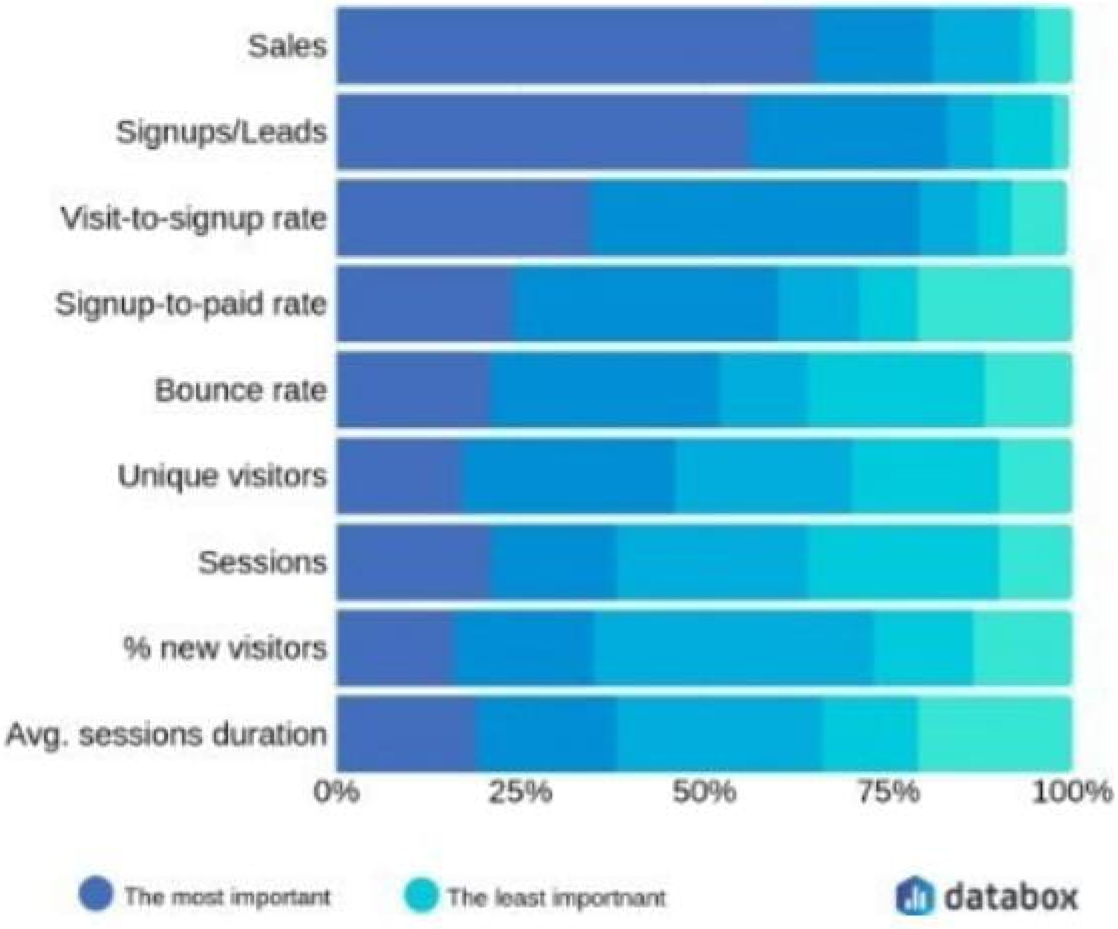
```
from flask import Flask,Response, render_template
fromcamera import Video


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

def gen(camera):
        while True:
        frame = camera get_frame()
        yield(b'__frame\r\n'
                    b'content- Type:image/jpeg\r\n\r\n\'+frame+
                    b'\r\n\r\n')
@app.route('/video_feed')
def video_feed():
         video = video()
          return Response(gen(video).mimetype='multipart/x-mixed-replace::boundary=frame')



if_name_ =='_main_':
          app.run()
```

## RESULTS

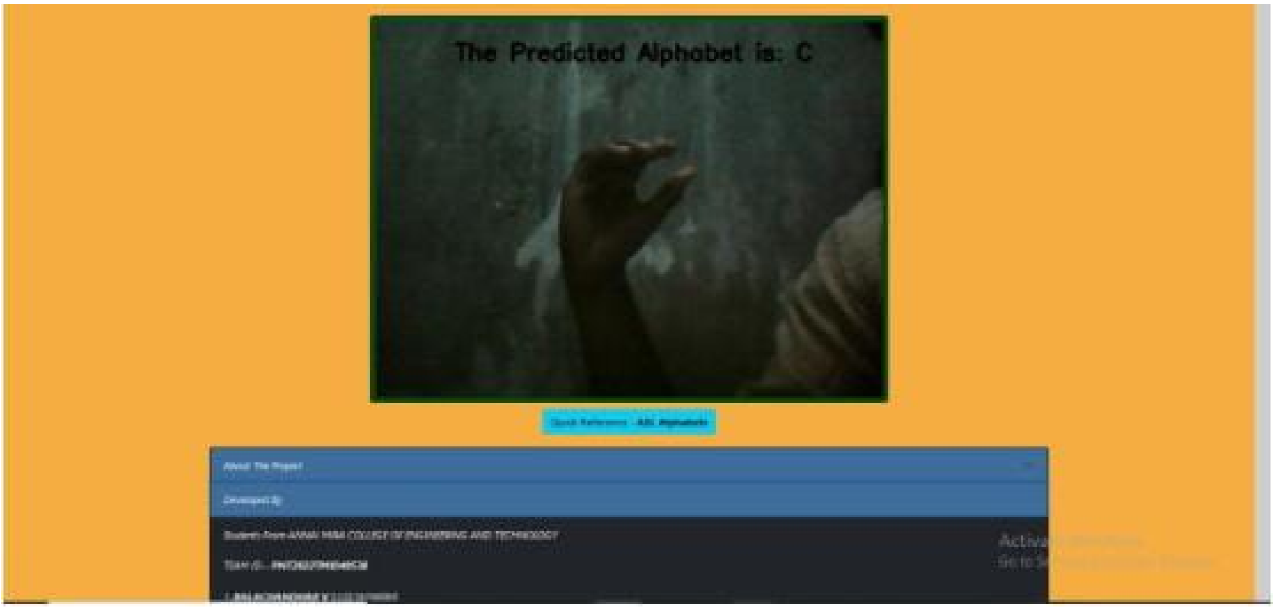
The proposedprocedure was implemented and tested on a set of images.

The trainingdatabase consists of 15750 imagesof Alphabets from "A" to "I", while the  testing database consists of 225 images of Alphabets from "A" to "I"

Once the gesture is recognized the equivalent alphabetis shown on the screen.



## OUTPUT



## ADVANTAGES AND DISADVANTAGES

### Advantages:

The speech is converted to sign language very quick to provide greater and fasterunderstanding to specially-abled people.

The user interface is convenient and simple for both people.

## Disadvantages:

The number of images and pixels for the model to train in the dataset is not high soaccuracy is moderate level.

It will be improved by changing the dataset.

Currently, we have deployed a dataset in the model for the alphabets A to I only.

## CONCLUSION:

It aims to bridge the communication gap between deaf mute people and the rest of society. The proposed methodology translates sign languageinto English alphabetsthat are understandable to humans. This system sends hand gestures to the model, who recognizes them and displaysthe equivalent.

## FUTURE SCOPE:

With the 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. 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 specially-abled people such as thosedeaf or dumb.

# IMAGE PREPROCESSING:

## image_dataset_from_directory

```
        tf.keras.preprocessing.image_dataset_from_directory(
  directory,
  labels="inferred",
  label_mode="int",
  class_names=None,
  color_mode="rgb",
  batch_size=32,
  image_size=(256, 256),
  shuffle=True,
  seed=None,
  validation_split=None,
  subset=None,
  interpolation="bilinear",
  follow_links=False,
  crop_to_aspect_ratio=False,
  **kwargs
)
```

**GENERATES A tf.data.Datasety from image files in a directory**

```
main_directory/
...class_a/
......a_image_1.jpg
......a_image_2.jpg
...class_b/
......b_image_1.jpg
......b_image_2.jpg
```

**load_image function**

```
tf.keras.preprocessing.image.load_img(
  path, grayscale=False, color_mode="rgb", target_size=None, interpolation="nearest"
)
```

**Loads an image into PIL FORMATE**

```
image = tf.keras.preprocessing.image.load_img(image_path)
input_arr = tf.keras.preprocessing.image.img_to_array(image)
input_arr = np.array([input_arr])  # Convert single image to a batch.
predictions = model.predict(input_arr)
```

**image_ to_ array funtion**

```
tf.keras.preprocessing.image.img_to_array(img,data_format=None, dtype=None)
```

**Converts a PIL image instance to a numpy array**

```
from PIL import Image
img_data = np.random.random(size=(100, 100, 3))
img = tf.keras.preprocessing.image.array_to_img(img_data)
array = tf.keras.preprocessing.image.img_to_array(img)
```

# MODEL BULIDIND:

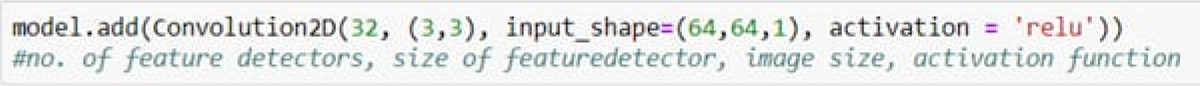
## Initialize The Mode:

Initialize the neural network layer by creating a reference/object to the Sequential class.

```
model=Sequential()
```

## Add The Convolution Layer

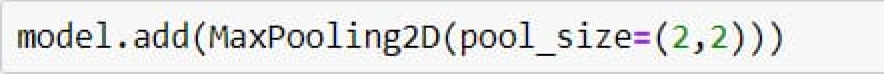
The first layer of the neural network model, the convolution layer will be added. To create a convolution layer, Convolution2D class is used. It takes the number of feature detectors, feature detector size, expected input shape of the image, activation function as arguments. This layer applies feature detectors on the input image and returns a feature map (features from the image).



```
model.add(Convolution2D(32, (3,3), input_shape=(64,64,1), activation = 'relu'))
#no. of feature detectors, size of featuredetector, image size, activation function
```

## Add The Pooling Layer

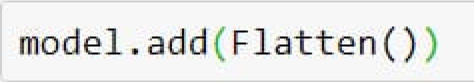
After the convolution layer, usually, the pooling layer is added. Max pooling layer can be added using MaxPooling2D class. It takes the pool size as a parameter. The efficient size of the pooling matrix is (2,2). It returns the pooled feature maps. (Note: Any number of convolution layers, pooling and dropout layers can be added)



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

## Add the flatten layer:

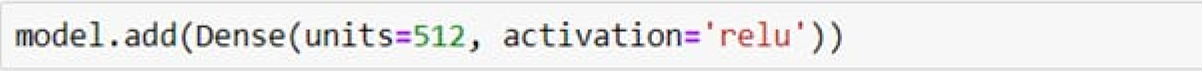
The flatten layer is used to convert the n-dimensional array to a 1-dimensional array. This 1D array will be given as input to ANN layers.



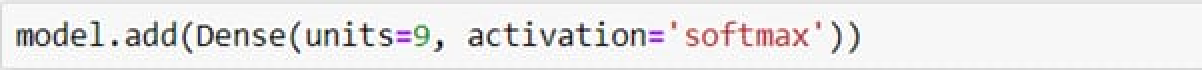
```
model.add(Flatten())
```

## Adding The Dense Layers

Three dense layers are added which usually takes the number of units/neurons. Specifying the activation function, kind of weight initialization is optional.



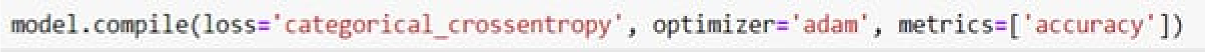
```
model.add(Dense(units=512, activation='relu'))
```



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

## Compile The Model

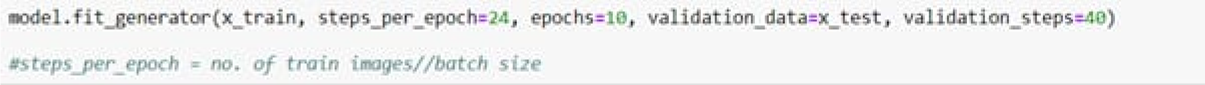
After adding all the required layers, the model is to be compiled. For this step, loss function, optimizer, and metrics for evaluation can be passed as arguments



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

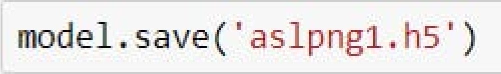
## Fit And Save The Model

Fit the neural network model with the train and test set, number of epochs, and validation steps.



```
model.fit_generator(x_train, steps_per_epoch=24, epochs=10, validation_data=x_test, validation_steps=40)
#steps_per_epoch = no. of train images//batch size
```

The weights are to be saved for future use. The weights are saved in as .h5 file using save().

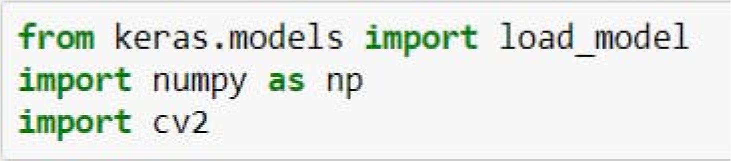


```
model.save('aslpng1.h5')
```

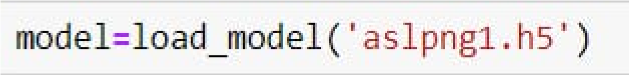
# TEST THE MODEL

## Import The Packages And Load The Saved Model

As a first step to start prediction we import packages that are used for loading the model and used to expand the dimension of the image. We use the Keras package to load the model which was saved when we built the model.



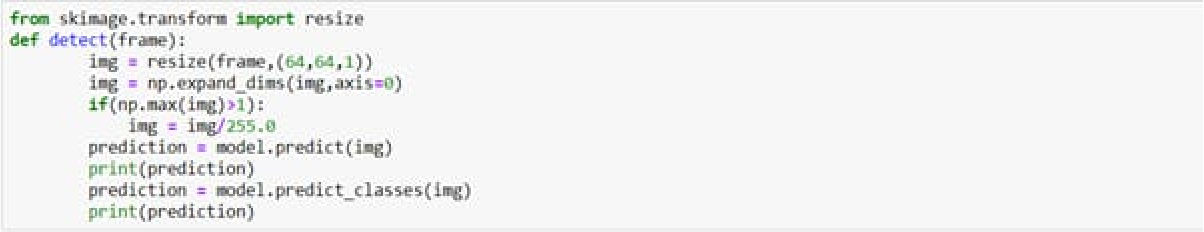
```
from keras.models import load_model
import numpy as np
import cv2
```



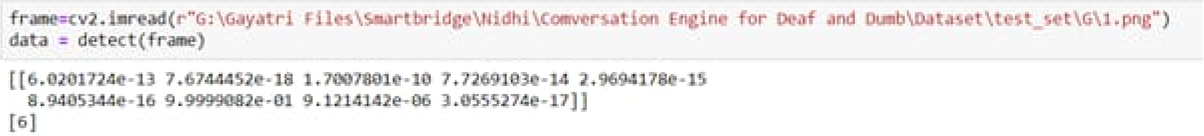
```
model=load_model('aslpng1.h5')
```

## Load The Test Image, Pre-Process It And Predict

Pre-processing the image includes converting the image to the array and resizing according to the model. Give the pre-processed image to the model to know to which class your model belongs to.



```
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):
            img = img/255.0
        prediction = model.predict(img)
        print(prediction)
        prediction = model.predict_classes(img)
        print(prediction)
```



```
frame=cv2.imread(r"G:\Gayatri Files\Smartbridge\Nidhi\Comversation Engine for Deaf and Dumb\Dataset\test_set\G\1.png")
data = detect(frame)
```

```
[[6.0201724e-13 7.6744452e-18 1.7007801e-10 7.7269103e-14 2.9694178e-15
  8.9405344e-16 9.9999082e-01 9.1214142e-06 3.0555274e-17]]
[6]
```

## APPLICATION BUILDING

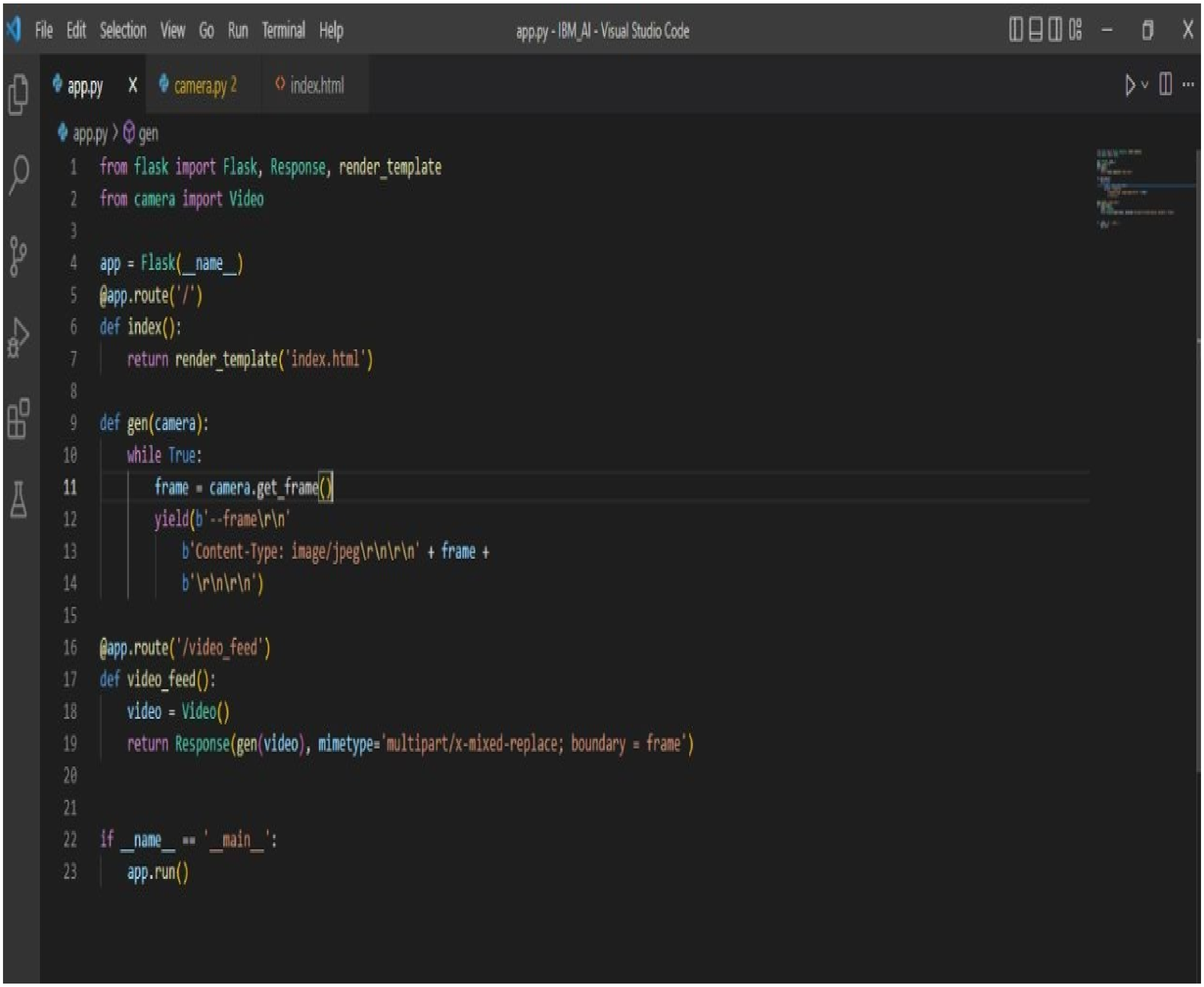
## BUILDING CAMERA:

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
class Video(object):
    def _init_(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        # self.model = load_model('asl_model.h5') # Execute Local Trained
```

```
Model
        self.model = load_model('IBM_Communication_Model.h5')  # Execute
IBM Trained Model
        self.index = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I']
        self.y = None
    def _del_(self):
        k = cv2.waitKey(1)
        self.video.release()
    def get_frame(self):
        ret, frame = self.video.read()
        frame = cv2.resize(frame, (64…
import cv2
video = cv2.VideoCapture(0)
while True:
        ret, frame = video.read()
        cv2.imshow("Frame", frame)
        k = cv2.waitKey(1)
        if k == ord('q'):
                break
video.release()
cv2.destroyAllWindows()
from flask import Flask, Response, render_template
from camera import video
app = Flask(name)
@app.router('/')
def index():
    return render_template('index.html')
def gen(camera):
    while True:
        frame = camera.get_frame()
        yield(b'--frame\r\n'
             b'content-Type: image/jpeg\r\n\n'+ frame +
              b'\r\n\r\n')
 @app.route('video_feed')
 def video_feed():
     video = video()
     return Response(gen(video), mimetype='multipart/x-mixed-replace;
boundary = frame')
 if name == 'main':
 app.run()
```

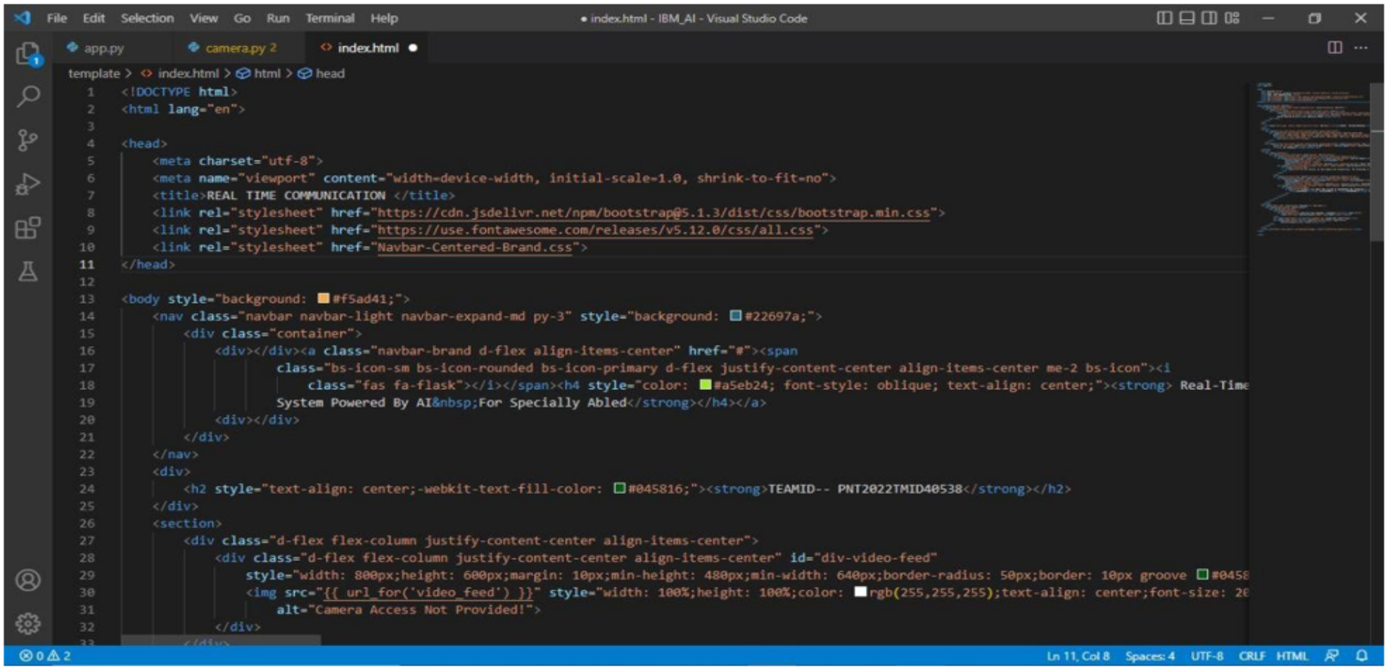
## APPENDIC

**SOURCE CODE:**
**FLASK:**

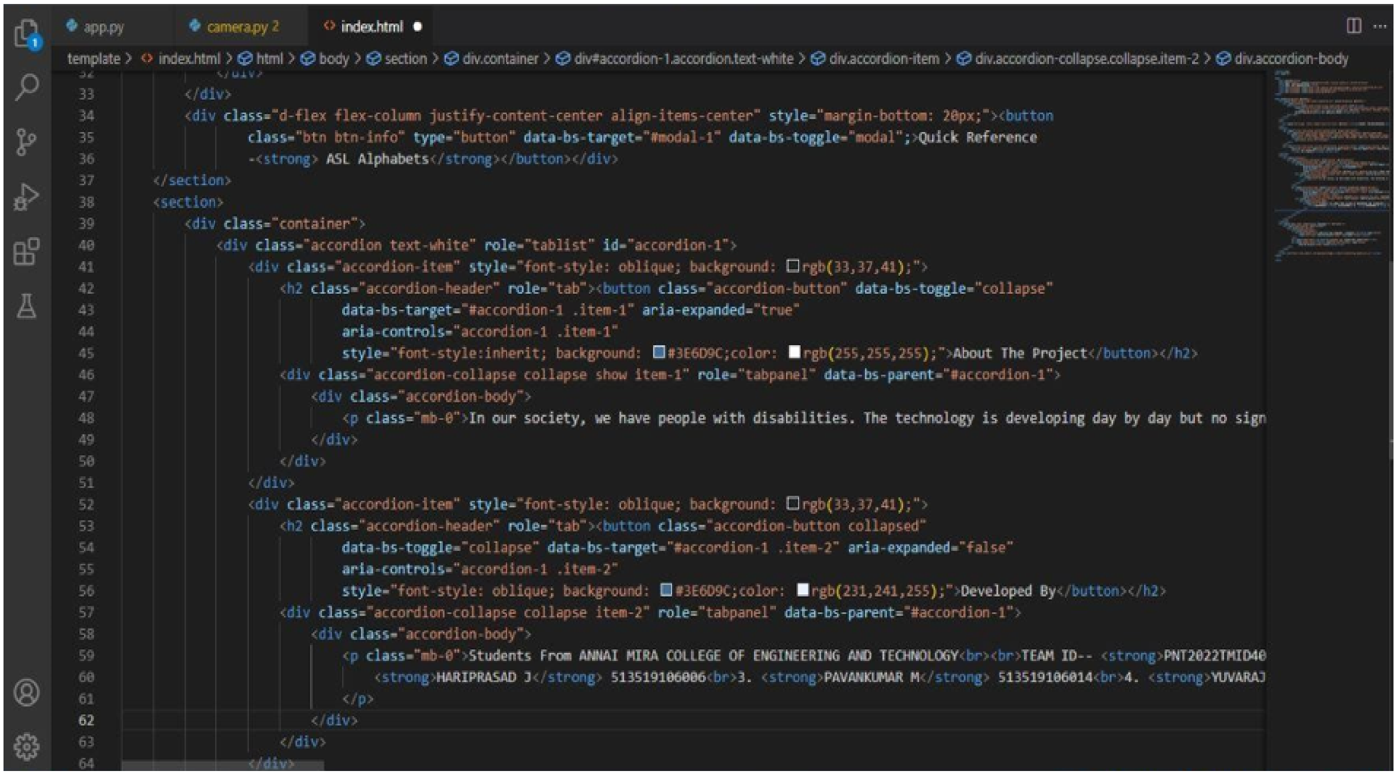


```python
from flask import Flask, Response, render_template
from camera import Video

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

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +
            b'\r\n\r\n')

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


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

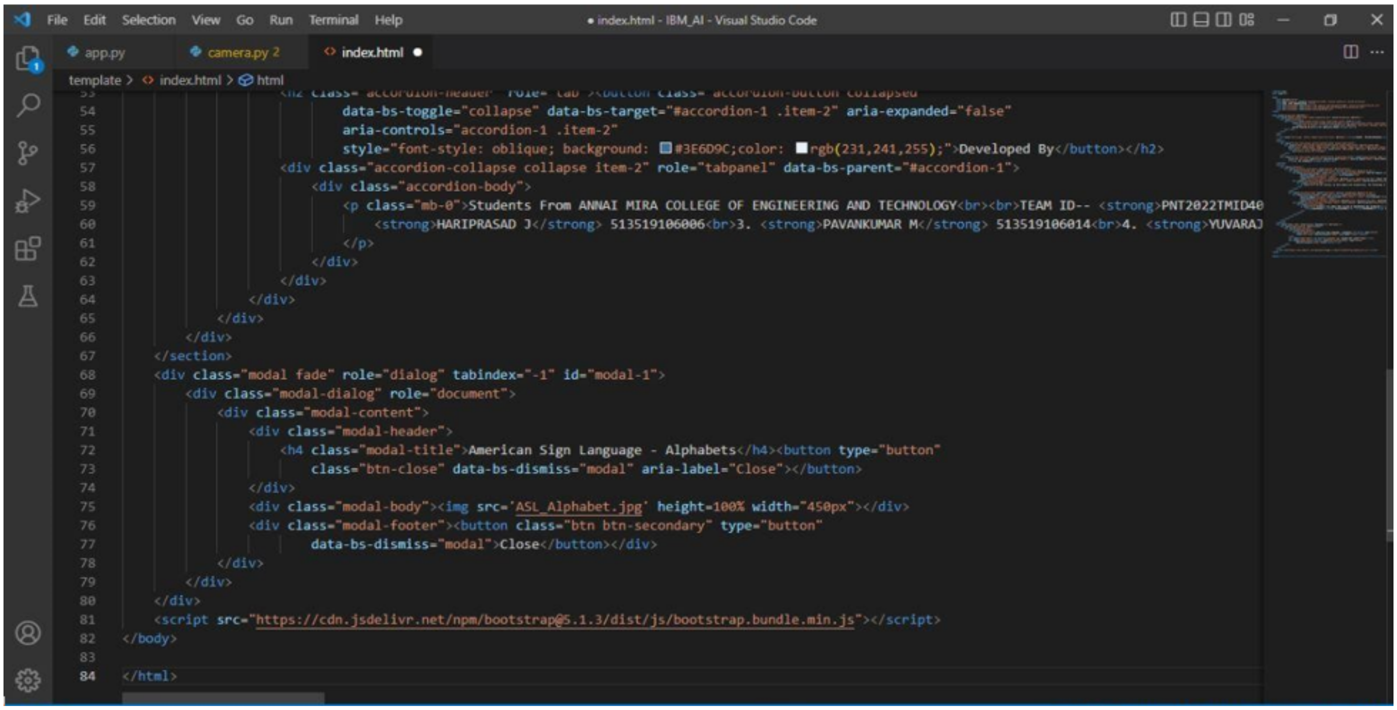
## HTML:



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>REAL TIME COMMUNICATION </title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.12.0/css/all.css">
    <link rel="stylesheet" href="Navbar-Centered-Brand.css">
</head>

<body style="background: #f5ad41;">
    <nav class="navbar navbar-light navbar-expand-md py-3" style="background: #22697a;">
        <div class="container">
            <div></div><a class="navbar-brand d-flex align-items-center" href="#"><span
                    class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex justify-content-center align-items-center me-2 bs-icon"><i
                        class="fas fa-flask"></i></span><h4 style="color: #a5eb24; font-style: oblique; text-align: center;"><strong> Real-Time
                    System Powered By AI&nbsp;For Specially Abled</strong></h4></a>
            <div></div>
        </div>
    </nav>
    <div>
        <h2 style="text-align: center;-webkit-text-fill-color: #045816;"><strong>TEAMID-- PNT2022TMID40538</strong></h2>
    </div>
    <section>
        <div class="d-flex flex-column justify-content-center align-items-center">
            <div class="d-flex flex-column justify-content-center align-items-center" id="div-video-feed"
                style="width: 800px;height: 600px;margin: 10px;min-height: 480px;min-width: 640px;border-radius: 50px;border: 10px groove #0458
                <img src="{{ url_for('video_feed') }}" style="width: 100%;height: 100%;color: rgb(255,255,255);text-align: center;font-size: 20
                    alt="Camera Access Not Provided!">
            </div>
```

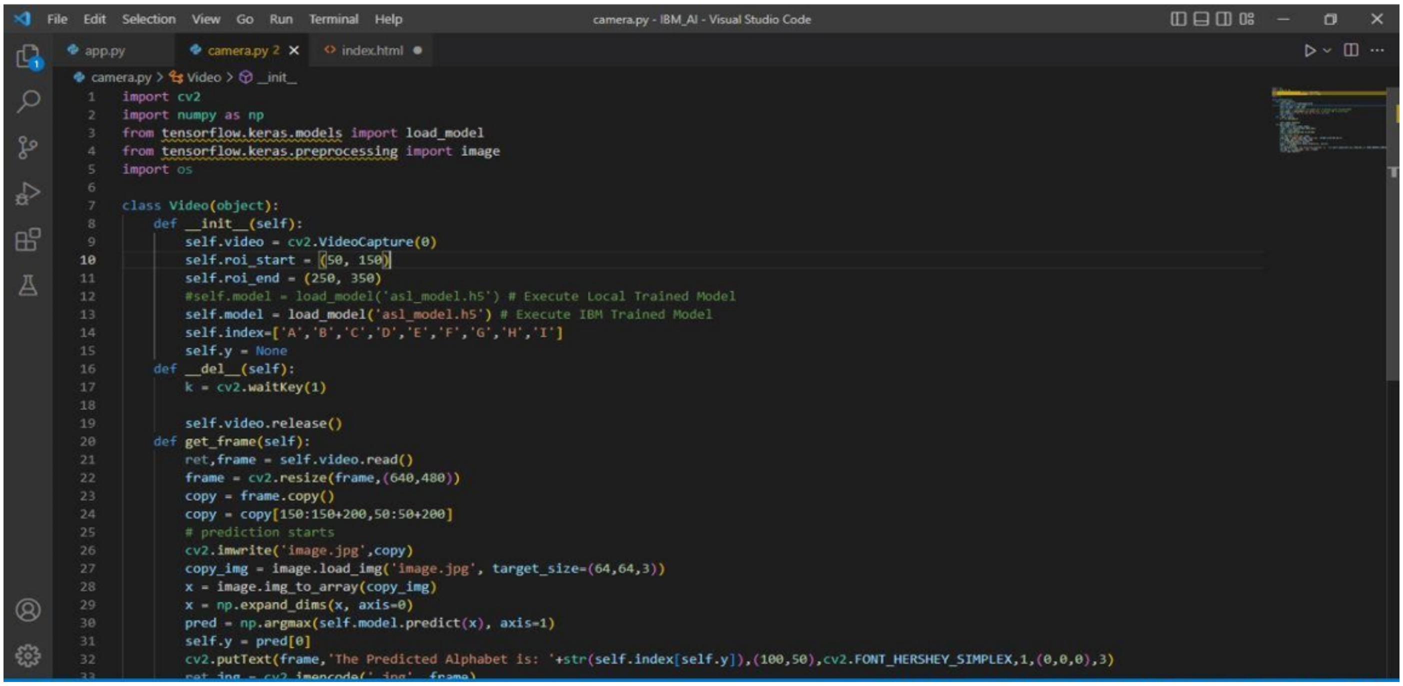

app.py
camera.py 2
index.html ●
template > index.html > html > body > section > div.container > div#accordion-1.accordion.text-white > div.accordion-item > div.accordion-collapse.collapse.item-2 > div.accordion-body

```html
        </div>
        <div class="d-flex flex-column justify-content-center align-items-center" style="margin-bottom: 20px;"><button
                class="btn btn-info" type="button" data-bs-target="#modal-1" data-bs-toggle="modal";>Quick Reference
                -<strong> ASL Alphabets</strong></button></div>
    </section>
    <section>
        <div class="container">
            <div class="accordion text-white" role="tablist" id="accordion-1">
                <div class="accordion-item" style="font-style: oblique; background: rgb(33,37,41);">
                    <h2 class="accordion-header" role="tab"><button class="accordion-button" data-bs-toggle="collapse"
                            data-bs-target="#accordion-1 .item-1" aria-expanded="true"
                            aria-controls="accordion-1 .item-1"
                            style="font-style:inherit; background: #3E6D9C;color: rgb(255,255,255);">About The Project</button></h2>
                    <div class="accordion-collapse collapse show item-1" role="tabpanel" data-bs-parent="#accordion-1">
                        <div class="accordion-body">
                            <p class="mb-0">In our society, we have people with disabilities. The technology is developing day by day but no sign
                        </div>
                    </div>
                </div>
                <div class="accordion-item" style="font-style: oblique; background: rgb(33,37,41);">
                    <h2 class="accordion-header" role="tab"><button class="accordion-button collapsed"
                            data-bs-toggle="collapse" data-bs-target="#accordion-1 .item-2" aria-expanded="false"
                            aria-controls="accordion-1 .item-2"
                            style="font-style: oblique; background: #3E6D9C;color: rgb(231,241,255);">Developed By</button></h2>
                    <div class="accordion-collapse collapse item-2" role="tabpanel" data-bs-parent="#accordion-1">
                        <div class="accordion-body">
                            <p class="mb-0">Students From ANNAI MIRA COLLEGE OF ENGINEERING AND TECHNOLOGY<br><br>TEAM ID-- <strong>PNT2022TMID40
                                <strong>HARIPRASAD J</strong> 513519106006<br>3. <strong>PAVANKUMAR M</strong> 513519106014<br>4. <strong>YUVARAJ
                            </p>
                        </div>
                    </div>
                </div>
```


File
Edit
Selection
View
Go
Run
Terminal
Help
• index.html - IBM_AI - Visual Studio Code
app.py
camera.py 2
index.html
template > index.html > html

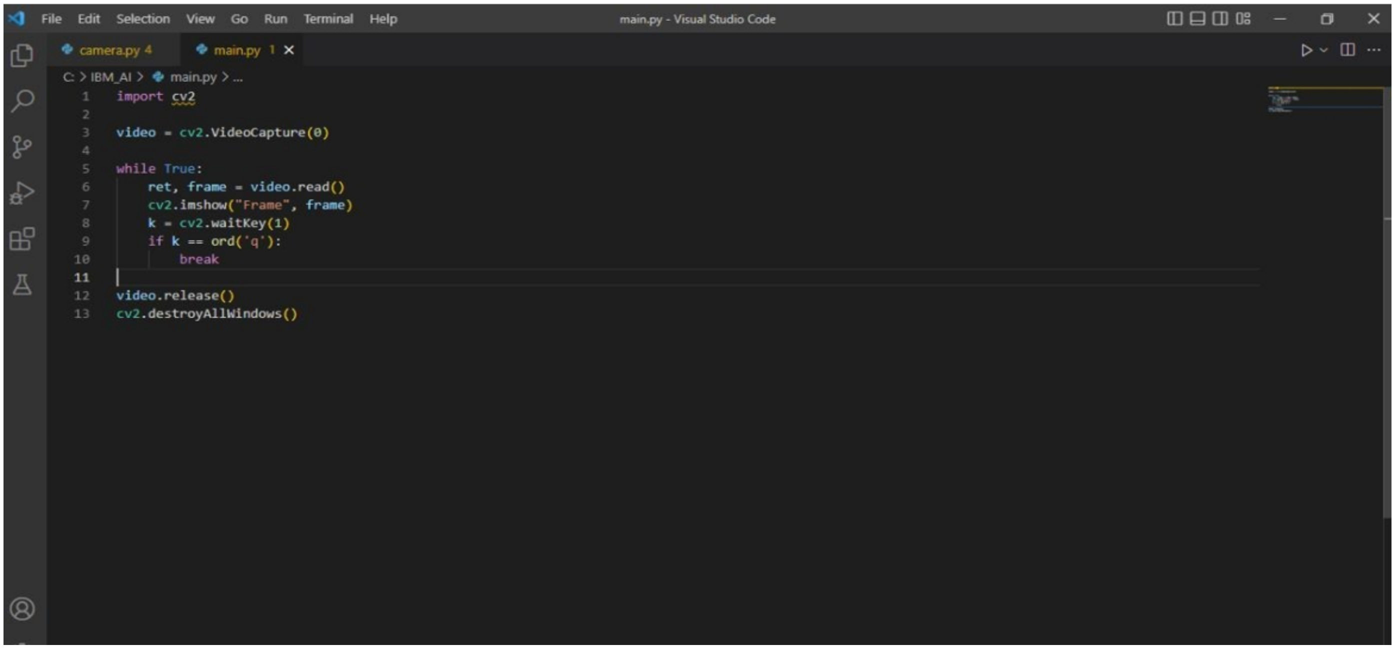
```html
            <h2 class="accordion-header" role="tab"><button class="accordion-button collapsed"
                    data-bs-toggle="collapse" data-bs-target="#accordion-1 .item-2" aria-expanded="false"
                    aria-controls="accordion-1 .item-2"
                    style="font-style: oblique; background: #3E6D9C;color: rgb(231,241,255);">Developed By</button></h2>
            <div class="accordion-collapse collapse item-2" role="tabpanel" data-bs-parent="#accordion-1">
                <div class="accordion-body">
                    <p class="mb-0">Students From ANNAI MIRA COLLEGE OF ENGINEERING AND TECHNOLOGY<br><br>TEAM ID-- <strong>PNT2022TMID40
                        <strong>HARIPRASAD J</strong> 513519106006<br>3. <strong>PAVANKUMAR M</strong> 513519106014<br>4. <strong>YUVARAJ
                    </p>
                </div>
            </div>
        </div>
    </div>
</div>
</section>
<div class="modal fade" role="dialog" tabindex="-1" id="modal-1">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h4 class="modal-title">American Sign Language - Alphabets</h4><button type="button"
                    class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body"><img src='ASL_Alphabet.jpg' height=100% width="450px"></div>
            <div class="modal-footer"><button class="btn btn-secondary" type="button"
                    data-bs-dismiss="modal">Close</button></div>
        </div>
    </div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>

</html>
```

## CAMERA:


File
Edit
Selection
View
Go
Run
Terminal
Help
camera.py - IBM_AI - Visual Studio Code
app.py
camera.py 2
index.html
camera.py > Video > __init__

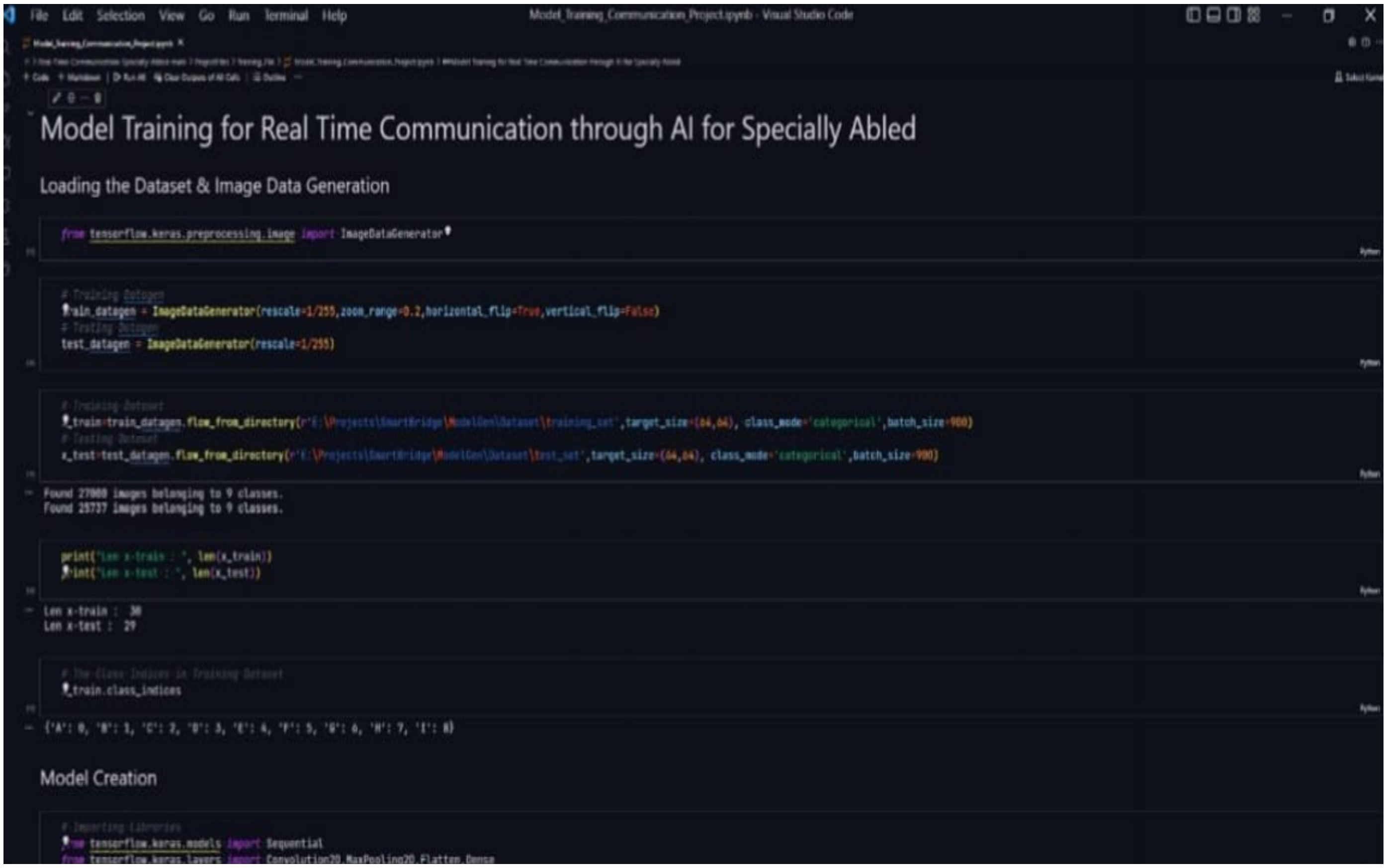
```python
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import os

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        #self.model = load_model('asl_model.h5') # Execute Local Trained Model
        self.model = load_model('asl_model.h5') # Execute IBM Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        k = cv2.waitKey(1)

        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame,(640,480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # prediction starts
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64,3))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
```

## MAIN:



```python
import cv2

video = cv2.VideoCapture(0)

while True:
    ret, frame = video.read()
    cv2.imshow("Frame", frame)
    k = cv2.waitKey(1)
    if k == ord('q'):
        break

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

## TRAINED MODEL:


File Edit Selection View Go Run Terminal Help
Model_Training_Communication_Project.ipynb - Visual Studio Code
Model_Training_Communication_Project.ipynb

# Model Training for Real Time Communication through AI for Specially Abled

## Loading the Dataset & Image Data Generation

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

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'E:\Projects\SmartBridge\ModelGen\Dataset\training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found 27000 images belonging to 9 classes.
Found 25737 images belonging to 9 classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train :  30
Len x-test :  29
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

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

## Model Creation

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