RMK ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601 206, Gummidipoondi Taluk, Thiruvallur District.

PROJECT

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

DONE BY

TEAM ID: PNT2022TMID15666

DINESH K (111719104044)

DINESH KUMAR M (111719104047)

BHARATH T (111719104022)

AJAY RAHUL V (111719104008)

SAHITH C H (111719104032)

CONTENTS

PROJECT REPORT

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

1.INTRODUCTION:

1.1 PROJECT OVERVIEW:

- Dumb people are usually face some problems on normal communication with other people in society. It has been observed that they sometimes find it difficult to interact with normal people with their gestures.
- Because people with hearing problems or deaf people cannot speak like normal people, they have to depend on a kind of visual communication in most cases.

1.2 PURPOSE:

- To overcome these problems, we have proposed a system that uses cameras to capture and convert videos of hand gestures from dumb people who turn into speech for understanding normal people.

2.LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

- Communication plays a significant role in making the world a better place. Most people communicate efficiently without any issues, but many cannot due to disability.
- They cannot hear or speak, which makes Earth a problematic place to live for them. Even simple basic tasks become difficult for them.

2.2 REFERENCES:

[1] Prof. P.G. Ahire, K.B. Tilekary,T.A. Jawake, P.B. Warale, “Two Way Communicator between Deaf and Dumb People and Normal People", 978-1-4799-6892-3/15 31.00 c 2015 IEEE.

[2] Shreyashi Narayan Sawant, "Sign Language recognition System to aid Deaf-dumb People Using PCA", IJCSET ISSN : 2229-3345 Vol. 5 No. 05 May 2014.

[3] Amitkumar Shinde, Ramesh Kagalkar, ”Sign Language to Text and Vice Versa Recognition using Computer Vision in Marathi”, International Journal of Computer Applications (0975 – 8887) National Conference on Advances in Computing (NCAC 2015) .

[4] Setiawardhana, Rizky Yuniar Hakkun, Achmad Baharuddin, "Sign Language Learning based on Android For Deaf and Speech Impaired People", 978-1-4673-9345- 4/15/31.00 c 2015 IEEE.

2.3 PROBLEM STATEMENT DEFINITIONS:

- The primary application for addressing the sign language is the improvement of the sign language. Computer recognition of the sign language is an important research problem for communication with the hearing impaired.
- The system proposed to develop and build an intelligent system that uses image processing, machine learning and artificial intelligence concepts to make visual inputs of hand gestures of sign language and to create an easily recognizable form of outputs.

3. IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS:

3.2 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	➢ Dumb people are usually face some problems on normal communicationwith other people in society. ➢ Our goal is to design a human computer interface system that can accurately identify the language of thedeaf and dumb.
2.	Idea / Solution description	➢ The system proposed to develop and build an intelligent system that uses image processing, machine learning and artificial intelligence concepts to make visual inputs of hand gestures.
3.	Novelty / Uniqueness	➢ When user will start recognition activity and give various hand gestures in front of camera, sign will be detected and speech will be produced to announce detected sign.
4.	Social Impact / Customer Satisfaction	➢ User will do different hand gestures infront of camera. ➢ User will able to see video, recognizedsign on GUI.
5.	Business Model (Revenue Model)	➢ First the application is tested with fewpeople. ➢ Further improvements can be done in the implementation of the communicator with other sign language such as American Sign Language, recognition of emotions in sign language and language Translation.
6.	Scalability of the Solution	➢ Hand gestures of deaf peoples by normal peoples this system is proposed. System gives output in the form of sound.

3.3 PROBLEM SOLUTION FIT:

1.CUSTOMER SEGMENT(S)

Normal People vs physically abled peoples

5.CUSTOMER CONSTRAINTS

1. Accuracy of system may vary depending upon light intensity changes.
2. Also accuracy depends upon distance between camera and object.

8.AVAILABLE SOLUTIONS

1. Systems interface will allow user to start video from camera.
2. User will do different hand gestures in front of camera.
3. User will able to see video, recognized sign on GUI.
4. User will get output in the form of sound which is converted from Speech of recognized sign.

2.JOBS-TO-BE-DONE / PROBLEMS

- Check the nature of languages.
- Check whether the language is understandable or not.
- Check whether the converted sign language is reliable or not.

6.PROBLEM ROOT CAUSE

1. Root cause Analysis supported by input from the problem-sufferers and improper maintenance are the main causes of this problem.

9.BEHAVIOUR

1. User will start video from camera.
2. User will be able to register different signs for further recognition using camera.
3. When user will start recognition activity and give various hand gestures in front of camera, sign will be detected and speech will be produced to announce detected sign

3.TRIGGERS

By using this application , users can avoid the unnecessary interaction with normal people.

4.EMOTIONS: BEFORE /AFTER

Before using this application , the user cannot understand the sign language of specially abled people.

By using this Application user knows how to communicate with specially abled people.

7. YOUR SOLUTION

1. The proposed communication system between Deaf and Dumb people and ordinary people are aiming for it when bridging the communication gap between two societies.
2. Several work is done earlier in this area, but this paper adds in complete two - sided communication in an efficient manner because the system is implemented as one Handy mobile application. So, it really serves its needs in all aspects. The above strategies prove to be efficient In terms of time and accuracy.

10.CHANNELS of BEHAVIORS

8.1 ONLINE

Online portal for making recognitions for a human interaction between normal people and specially abled people by using machine learning.

8.2 OFFLINE

During offline , some simple tests that can be done at home will be suggested so that users can find themself if necessary.

4. REQUIREMENTS ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	**LOW VISION:** As a user who has trouble reading due to low vision, I want to be able to make the text larger on thescreen so that I can read it. **Registration through Gmail**
FR-2	User Confirmation	**IMPAIRED USER:** As a user who is hearing -impaired, I want a turn on video captions so that I can understand what is beingsaid in videos. **Confirmation via Email**
FR-3	User Registration	**COLOR BLINDNESS**: As a user who is color blind, I want to links to bedistinguishable on the page so that I can find the links and navigate the site. **Registration through Gmail**

4.2 NON-FUNCTIONAL REQUIREMENTS:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• Visual and Audio Help • Text size scaling • Reverse contrast
NFR-2	**Security**	Important information: • Walking in single file or in narrow space. • Steps, Stairs and Slope. • Kerbs and Roads.
NFR-3	**Reliability**	To determine reliability measures are: • Test-Retest Repeatability • Individual Repeatability
NFR-4	**Performance**	To determine predictors of success in reading with low vision aids, in terms of reading acuity, optimum

		acuity reserve, and maximum reading speed, for observers with low vision for various causes.
NFR-5	**Availability**	Lack of adequate low vision services and barriers to their provision and uptake impact negatively on efforts to prevent visual impairment and blindness.
NFR-6	**Scalability**	There is a large selection of device to help people with low vision. Some are “Optical”, glass lenses such as magnifying glasses and telescopes.

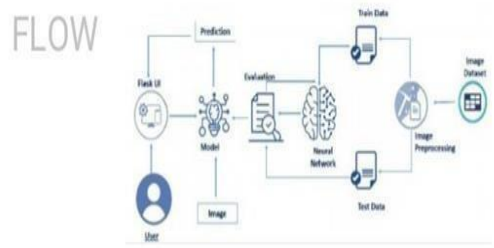
5.PROJECT DESIGN:

5.1 DATA FLOW DIAGRAMS:

Data Flow Diagrams:

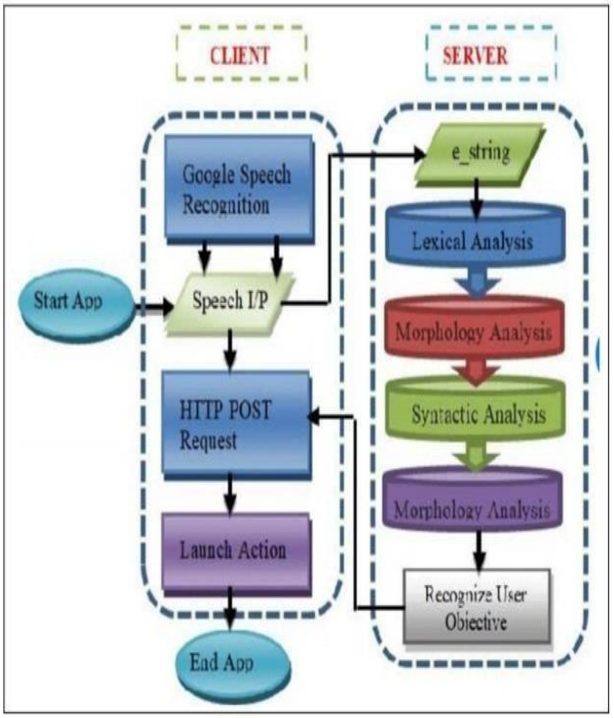
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example: (Simplified)



(i) Lexical analysis converts character sequence into token sequence.
(ii) Morphology analysis defines, analyzes, and describes the structure of language units of a particular language.
(iii) Syntactic analysis analyzes the text made from a series of markers to determine grammar structures.
(iv) Semantic Analysis relates syntactic structures from the levels of phrases and sentences to their language-independent meanings.

Example: Real time communication



5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Real-Time Communication System Powered by AI for Specially Abled

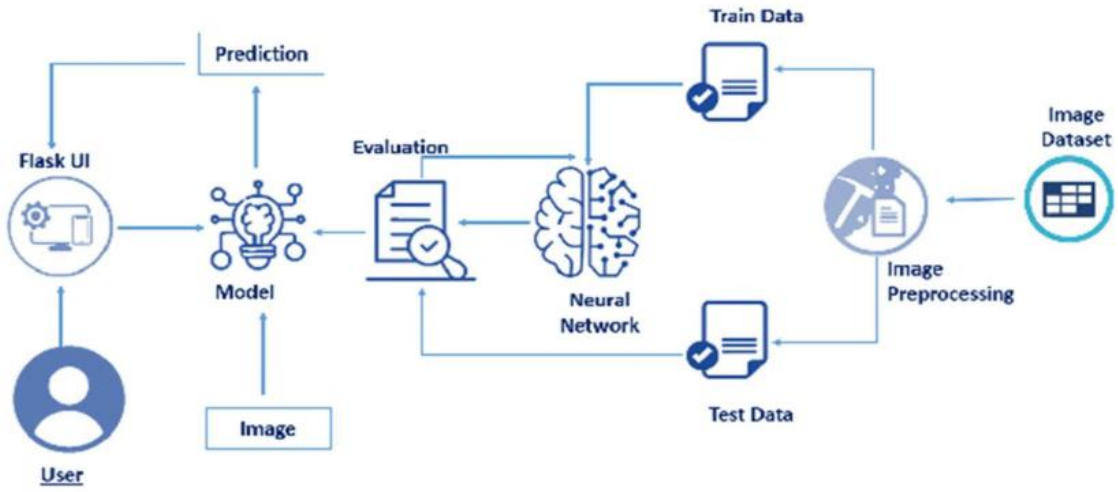


Table 1 : Components and Technologies

	Component	Description	Technology
1.	User	Deaf and dumb people willing to communicate efficiently, without any hassle with others in their surrounding environment	AI techniques
2.	Flask UI	The components of Flask's User Interface allow one to interact with clients that make use of your application and gather information.	Can be executed using existing cloudtechnologies
3.	Image Dataset	The initial prototype of this application is trained on a subset of the dataset containing 20 different signs adhering to the American Sign Language	AI techniques
4.	Image Preprocessing	The images in the dataset are preprocessed to increase the sharpness / clarity and remove any noise	ANN, CNN, OpenCV
5.	Training	SVM is run on the training dataset to extract attributes from the images which are then fed to the Neural Network in order to make the prediction	Scikit-learn, Natural Language Processing(NLP)

6.	Testing	The trained model is then run on an additional untested 10-15 sign-language images and the performance parameters are evaluated and recorded	Scikit-learn, NLP
7.	Neural Network	The same neural network architecture is used for both top-view and bottom-view models; the onlydifference lies in the number of output units	ANN
8.	Evaluation	Records the generalization accuracy of theproposed model on future / unseen data	
9.	Model	ML algorithms like SVM (Support Vector Machine) are applied to classify the given image dataset	Machine Learning
10.	Prediction	The attributes extracted from the images are examined and predictions are made in order to convert the sign-language to the correspondingtext	ANN, CNN

Table 2 :Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Robots and various other AI tools have made it possible for people with disabilities to live comfortably	AI techniques like self-moving robots and other software systems
2.	Security Implementations	Users are authenticated based on their username/password pair and/or OTP sent to theirgiven mobile numbers	SHA-1, Encryptions, IAM Controls
3.	Scalable Architecture	We implement a modular 3-tier client-server application architecture that improves scalability, availability, and performance. Individual tiers arecontainerized	Presentation layer, Application layer and Data Layer modularity, Docker
4.	Availability	The application has an extremely low downtimeand load balancers forward request to other available machines in case of failures	Key performance indicators (KPI)
5.	Performance	The application performs efficiently under a heavy load of translation requests without any significant reduction in the conversion accuracy	Number of requests per minute, accuracy of translation (sign-language to speech & text to sign-language)

5.3 USER STORIES:

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Deaf people)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can registered my account.	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can received confirmation gmail & click confirm.	High	Sprint-2
	Login	USN-3	As a user, I can log into the application through gmail .	I can registered & access the dashboard with gmail Login.	Low	Sprint-1
		USN-4	As a user, I can see my application and made changes in any browsers.	I can login and see my Account at anywhere.	Medium	Sprint-2
	Dashboard	USN-5	As a user, I can create my account in a given dashboard.	I can access my account / Dashboard.	High	Sprint-1
Customer (Dumb people)	Registration	USN-6	As a User, I can regiser my application through gmail.	I can registered my account .	High	Sprint-2
		USN-7	As a User, I can receive confirmation mail and get verification code from OTP and gmail.	I can received confirmation mail & click confirm.	Low	Sprint-2
	Login	USN-8	As a User, I can log into my account by any web browsers.	I can login and see my account.	Medium	Sprint-1
	Dashboard	USN-9	As a User, I can create my account in a given Dashboard.	I can created my account & access into dashboard.	High	Sprint-2

6. PROJECT PLANNING AND SCHEDULING:

6.1 SPRINT PLANNING AND ESTIMATION:

S.No	Milestone	Activities	Team Members
1.	Data Collection	Create Train and Test Folders	Dinesh K Bharath T Dinesh Kumar M
2.	Image Preprocessing	Import Image Data Generator Library and Configureit	Ajay Rahul Sahith
3.	Image Preprocessing	Apply Image Data Generat or functionality to Train and Test set	Dinesh K Bharath T Dinesh Kumar M

4.	Model Building	Import the required model building libraries	Ajay Rahul Sahith
5.	Model Building	Initialize the model	Dinesh K Bharath T Dinesh Kumar M
6.	Model Building	Add the convolution layer	Dinesh K Ajay Rahul
7.	Model Building	Add the pooling layer	Bharath T Sahith
8.	Model Building	Add the flatten layer	Dinesh Kumar M Ajay Rahul
9.	Model Building	Adding the dense layers	Dinesh K Bharath T Dinesh Kumar M
10.	Model Building	Compile the model	Bharath T Dinesh Kumar M
11.	Model Building	Fit and save the model	Dinesh K Bharath T
12	Test the Model	Import the packages and load the saved models	Dinesh Kumar M Ajay Rahul Sahith
13.	Test the model	Load the test image, pre-process it and predict	Dinesh K Bharath T Dinesh Kumar M
14.	Application Building	Build a flask application	Dinesh K Dinesh Kumar M
15.	Application Building	Build the HTML page	Bharath T Dinesh Kumar M Sahith
16.	Application Building	Output	Bharath T Dinesh Kumar M Sahith
17.	Train CNN Model on IBM	Register for IBM Cloud	Dinesh K Dinesh Kumar M Bharath T
18.	Train CNN Model on IBM	Train Image Classification Model	Dinesh K Dinesh Kumar M Bharath T

6.2 SPRINT DELIVARABLE SCHEDULE:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Dinesh K Bharath T
Sprint-2		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Dinesh Kumar M
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	1	Medium	Sahith Challa
Sprint-2	Dashboard	USN-4	As a user, I can log into my account in a given Dashboard	1	High	Ajay Rahul
Sprint-1	User interface	USN-5	Professional responsible for user requirements & needs	1	High	Dinesh K
Sprint-3	Objective	USN-6	The goal is to describe all the inputs and outputs	1	High	Bharath T
Sprint-4	Privacy	USN-7	The developed application should be secure for the users	1	High	Dinesh Kumar M

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING & SOLUTIONING

7.1 Feature 1

Skin colour recognition: Skin detection is the process of finding skin coloured pixels and regions in an image or video. This process is typically used as a pre processing step to find areas that may have human faces and limbs in images.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
# Create a image

img1 = np.zeros((400,600,3),np.uint8)
plt.imshow(img1)
# Draw a circle

circle = cv2.circle(img1, (300,200), 50, (255,0,0), -1)   # (0,0,0)--->(R,G,B)
plt.imshow(img1)
# Drawing rectangle

rectangle = cv2.rectangle(img1,(200,100),(400,300),(0,255,0),6)
plt.imshow(img1)
# Drawing line

line1 = cv2.line(img1,(200,100),(400,300),(0,0,255),4)
line2 = cv2.line(img1,(200,300),(400,100),(0,0,255),4)
plt.imshow(img1)
circle = cv2.circle(img1, (300,200), 50, (255,255,0), -1)   # (0,0,0)---
>(R,G,B)
plt.imshow(img1)
# Text on image

text = cv2.putText(img1, 'openCV', (200,50), cv2.FONT_HERSHEY_SIMPLEX, 2,
(255,255,255),5)
plt.imshow(img1)
# Reading the image

img = cv2.imread('boy.jpg',1)
plt.imshow(img)
# Convert BGR to RGB

img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img_rgb)
```

7.2 Feature 2

Template Matching: Template matching is a technique in digital image processing to find small portions of an image that match a template image. It can be used in manufacturing as part of quality control, one way to navigate a mobile robot, or as a way to detect edges in images.

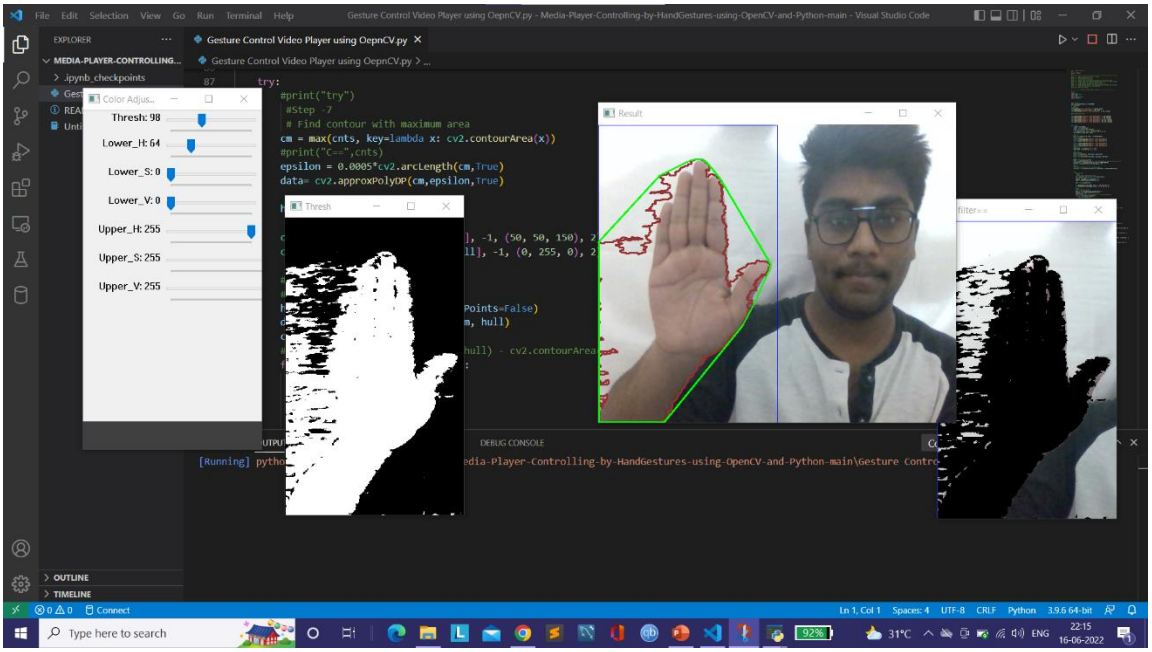
```
# Convert BGR to RGB

img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img_rgb)
# Convert BGR to Gray

img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(img_gray)
# Finding shape

img_rgb.shape
img_gray.shape
# Resize the image

resize = cv2.resize(img_rgb,(500,1000))
print(resize.shape)
plt.imshow(resize)
# Image crop

crop = resize[130:370,150:300]
plt.imshow(crop)
# Edge Detection

edge = cv2.Canny(img_rgb,100,200)
plt.imshow(edge)
# Blur image

r = resize[130:370,150:300]
blur = cv2.GaussianBlur(r,(13,13),cv2.BORDER_DEFAULT)
plt.imshow(resize)
plt.imshow(blur)
```

8. TESTING

8.1 Test Cases

It’s a Desktop application.

- User will start video from camera.
- User will be able to register different signs for further recognition using camera.
- When user will start recognition activity and give various hand gestures in front of camera, sign will be detected and speech will be produced to announce detected sign.



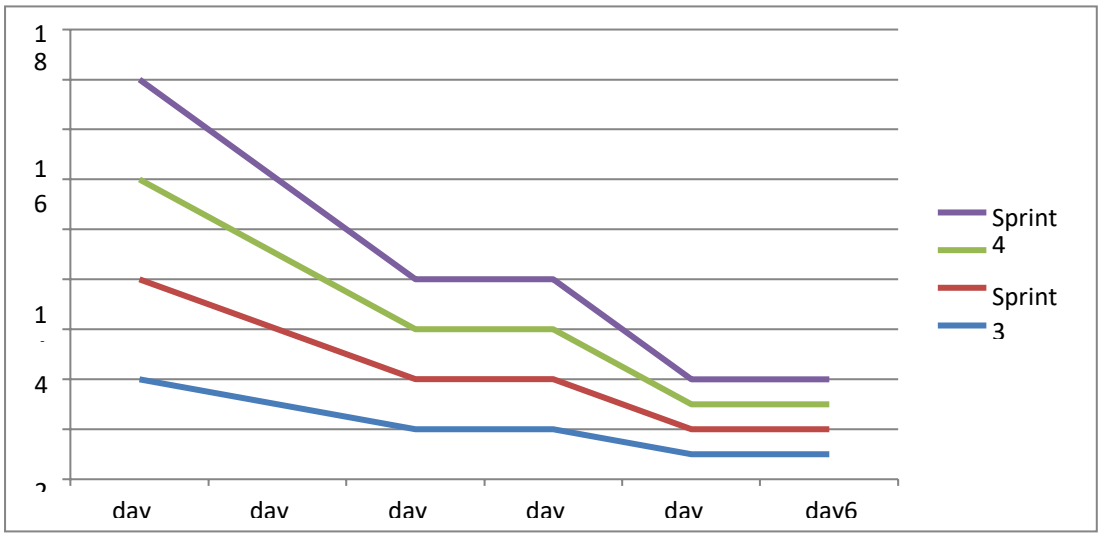
8.2 User Acceptance Testing

- Systems interface will allow user to start video from camera.
- User will do different hand gestures in front of camera.
- User will able to see video, recognized sign on GUI.
- User will get output in the form of sound which is converted from Speech of recognized sign.

9. RESULTS

9.1 Performance Metrices

- Accuracy of system may vary depending upon light intensity changes.
- Also accuracy depends upon distance between camera and object.



10. ADVANTAGES AND DISADVANTAGES

Advantages:

- Real Time Communication Saves time.
- Real Time Communication Increases Productivity.
- Communication can be 24/7.
- Easy to setup and communicate.
- Quick decisions can be made.

Disadvantages:

- Users who are not aware of tech finds challenging.
- The chances of misunderstanding are high.
- Scheduling can be challenge.

11. CONCLUSION

The proposed communication system between Deaf and Dumb people and ordinary people are aiming for it when bridging the communication gap between two societies. Several work is done earlier in this area, but this paper adds in complete two - sided communication in an efficient manner because the system is implemented as one Handy mobile application. So, it really serves its needs in all aspects. The above strategies prove to be efficient In terms of time and accuracy. Further improvements can be done in the implementation of the communicator with other sign language such as American Sign Language, Accent recognition for different accents throughout Globe, recognition of emotions in sign language and language Translation.

12. FUTURE SCOPE

- Proposed systems scope is related with education of dumb peoples. Dumb people faces many problems when normal person could not understand their language. They were facing communication gap with normal peoples.
- For communication between deaf person and a second person, a mediator is required to translate sign language of deaf person. But a mediator is required to know the sign language used by deaf person. But this is not always possible since there are multiple sign languages for multiple languages.
- So to understand all sign languages, Hand gestures of deaf peoples by normal peoples this system is proposed. System gives output in the form of sound.

13. APPENDIX

Source Code

##IMPORT LIBRARIES

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D

from tensorflow.keras.preprocessing.image import ImageDataGenerator

import numpy as np

import matplotlib.pyplot as plt

import cv2
```

##UNZIP THE FILE

```
!unzip '/content/drive/MyDrive/IBMPROJECT/conversation engine for deaf and dumb.zip'
```

##DATA AUGMENTATION

from keras.preprocessing.image import ImageDataGenerator

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

test_datagen = ImageDataGenerator(rescale=1./255)

x_train = train_datagen.flow_from_directory("../DATA COLLECTION/training_set", target_size=(64,64),batch_size=100,class_mode='categorical', color_mode ="grayscale")

x_test = test_datagen.flow_from_directory("../DATA COLLECTION/test_set", target_size=(64,64),batch_size=100,class_mode='categorical', color_mode ="grayscale")

len(x_train)

len(x_test)

x_train.class_indices

##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

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)

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('Real_time.h5')

img = image.load_img('../DATA COLLECTION/test_set/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("../DATA COLLECTION/test_set/test_set/H/107.png")

detect(img)

img = image.load_img('../DATA COLLECTION/test_set/test_set/A/110.png')

pred=detect(img)

img=image.load_img('../DATA COLLECTION/test_set/test_set/E/111.png')

detect(img)
```

##DEPLOY CODE

```
import cv2 #mporting opencv Library this i to open camera and take the video
import numpy as np # to convert image to array and expand dimensions
from tensorflow.keras.models import load_model # to Load the saved model
from tensorflow.keras.preprocessing import image # to preproccess the image
model = load_model("asl_model_84_54.h5") # we are loading the saved moodek
video = cv2.VideoCapture(0) # two parameters 1, bool 0 or 1, frame
index = ["A","B","C","D","E","F","G","H","I"]
index=['A','B','C','D','E','F','G','H','I']
#from playsound import playsound
while(1):
   success,frame = video.read()
   cv2.imwrite("boy.jpg",frame)
   img = image.load_img("boy.jpg",target_size = (64,64))
   x = image.img_to_array(img)
   x = np.expand_dims (x,axis = 0)
   pred = np.argmax(model.predict(x),axis=1)
   p = index [pred[0]]
   print("predicted letter is: "+ str(p))
   #playSound("letter"+str(str(index [p])+"is detected"))
   cv2.putText (frame, "predicted letter is "+str(p), (100, 100), cv2.
FONT_HERSHEY_SIMPLEX, 1,(0,0,0), 4)
   cv2.imshow("showcasewindow", frame)
   if cv2.waitkey(1) & 0xFF == ord('a'):  break
video.release()
cv2.destroyAllwindows()
```

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-28482-1660112776

Project Demo Link

https://drive.google.com/drive/folders/1XGCvEq4yBTikIKOfdpa39_fY0Mc6wLfT?usp=share_link