Final Report

A Real Time Communication System For Specially Abled

Team Members:

MOHAMED IBURAHIM.P

MOHITH.S

NANDA KISHORE.K.P

NITISH KUMAR.N

Table Of Contents

1. INTRODUCTION

1.1 Project Overview :

People with impairments are a part of our society. Though technology is constantly evolving, little is being done to improve the lives of these people. It has never been easy to communicate with a deaf-mute individual. It is quite challenging for silent persons to communicate with non-mute people, mostly because hand sign language is not taught to the general public. It might be quite challenging for them to communicate at times of crisis.In circumstances where other modes of communication, like speech, cannot be employed, the human hand has continued to be a preferred choice. A proper discussion between a normal person and an impaired person in any language will be made possible by a voice conversion system with hand gesture recognition and translation. The project intends to create a system that can translate speech into acceptable sign language for the deaf and dumb as well as translate sign language into a human hearing voice in the desired language to communicate a message to normal people. A convolution neural network is being used to build a model that is trained on various hand motions. On the basis of this model, an app is created. With the help of this app, persons who are deaf or dumb can communicate using signs that are translated into speech and human-understandable words.

1.2 Purpose :

The only means of communication for someone who cannot hear or speak is through sign language. People who are physically disabled can communicate their thoughts and feelings more effectively by using sign language.The alphabets and motions of sign language have been identified in this work using a revolutionary system of sign language recognition. We can identify the indications and provide the appropriate text output with the use of computer vision and neural networks.

2. LITERATURE SURVEY

2.1 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. Communications between deaf-mute and a normal person has always been a challenging task. It is complicated 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 handy to have a proper conversation between a normal person and an impaired person in any language.

2.2 References

Design of Communication Interpreter for Deaf and Dumb Person was published by Pallavi Verma (Electrical and Electronics Department, Amity University, Greater Noida, Uttar Pradesh, India), Shimi S. L (Assistant Professor, NITTTR, Chandigarh, India), Richa Priyadarshani (Electrical and Electronics Department, Amity University, Greater Noida, Uttar Pradesh, India).International Journal of Science and Research (IJSR) · Jan 2013

Development of full duplex intelligent communication system for deaf and dumb
people was published in the year January 2017
DOI:10.1109/CONFLUENCE.2017.7943247

At 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence (Confluence) by Surbhi Rathi Department of Information Technology, Yeshwantrao Chavan College of Engineering Nagpur, India and Ujwalla Gawande, Department of Information Technology Yeshwantrao Chavan College of Engineering Nagpur, India.

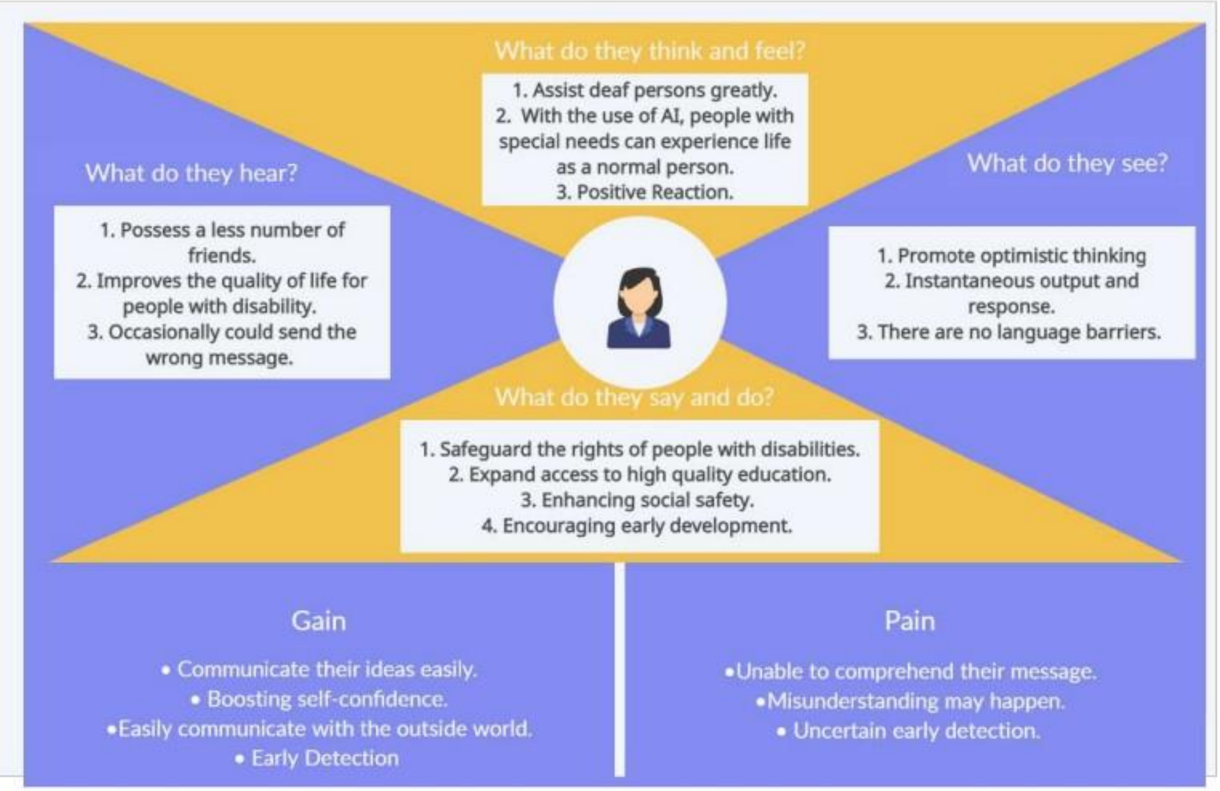
A Review Paper on Sign Language Recognition for The Deaf and Dumb published by R Rumana(B.E Graduate(IV year), Department of Computer Science and Engineering, SCSVMV, Kanchipuram) , Reddygari Sandhya Rani(B.E Graduate(IV year), Department of Computer Science and Engineering, SCSVMV, Kanchipuram) , Mrs. R. Prema(Assistant Professor, Department of Computer Science and Engineering, SCSVMV, Kanchipuram).Published (First Online): 01-11-2021

2.3 Problem Definition Statement

People with disabilities are a part of our society. Even though technology is constantly evolving, little is being done to improve the lives of these people. Communication with a deaf-mute person has always been difficult. Because hand sign language is not taught to the general public, it can be difficult for silent people to communicate with non-mute people. In times of crisis, they may find it difficult to communicate. When other modes of communication, such as speech, are unavailable, the human hand has remained a popular method of information transmission. A voice conversion system with hand gesture recognition and translation will be very helpful in establishing proper communication between a normal person and a handicapped person in any language.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



What do they think and feel?

1. Assist deaf persons greatly.
2. With the use of AI, people with special needs can experience life as a normal person.
3. Positive Reaction.

What do they hear?

1. Possess a less number of friends.
2. Improves the quality of life for people with disability.
3. Occasionally could send the wrong message.

What do they see?

1. Promote optimistic thinking
2. Instantaneous output and response.
3. There are no language barriers.

What do they say and do?

1. Safeguard the rights of people with disabilities.
2. Expand access to high quality education.
3. Enhancing social safety.
4. Encouraging early development.

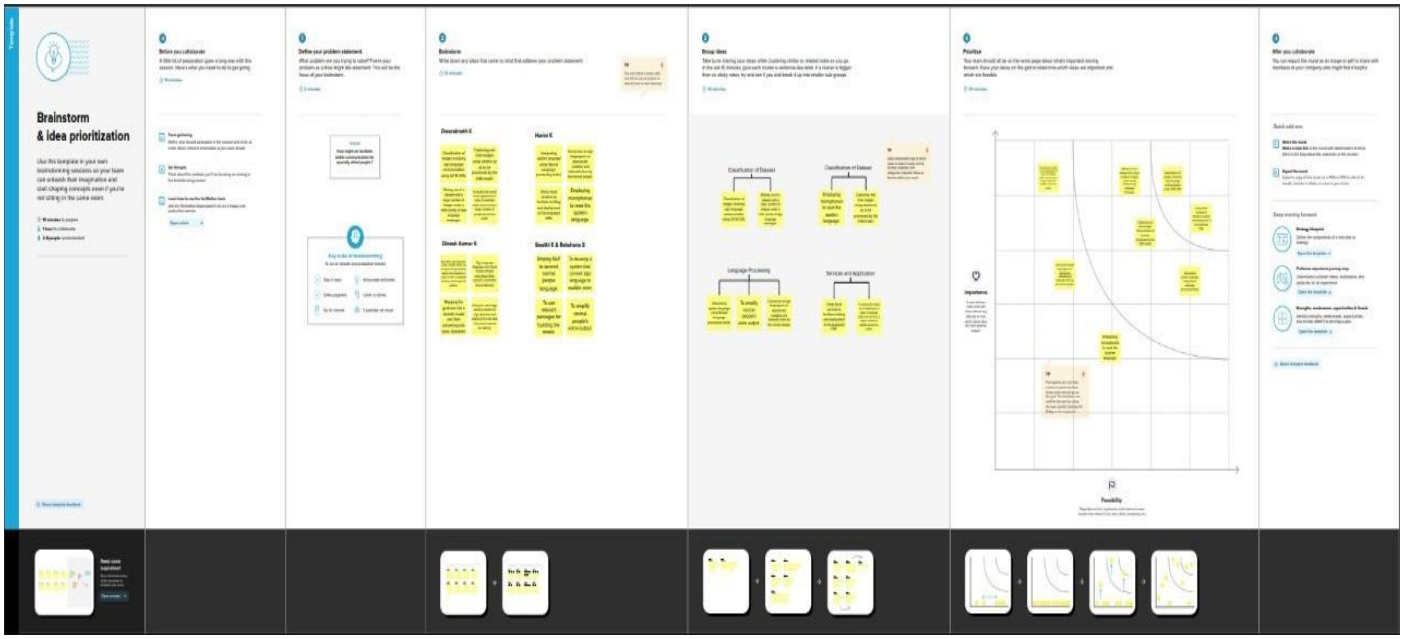
Gain

- Communicate their ideas easily.
- Boosting self-confidence.
- Easily communicate with the outside world.
- Early Detection

Pain

- Unable to comprehend their message.
- Misunderstanding may happen.
- Uncertain early detection.

3.2 Ideation & Brainstorming

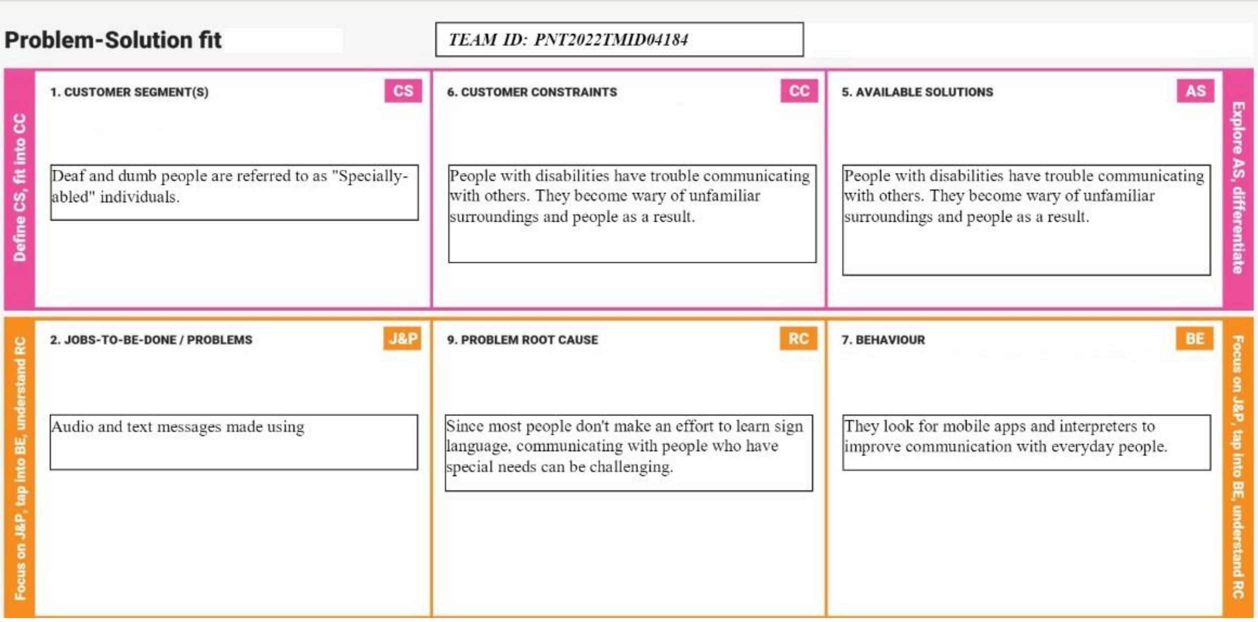


3.3 Proposed Solution

The project's goal is to develop a system that will help deaf and dumb people communicate in real time. The model uses background-subtraction, a method that is thought to be a crucial pre-processing step. These methods will be implemented in the development of the proposed Sign Language Converter in order to locate the hand region in the input image captured by the camera.

We'll be using the most latest and well-liked wearable technology, which allows the disabled person to effortlessly take the gadget (a mobile application) wherever and everywhere, facilitating communication for both people with special needs and regular people. The performance of the trained model will be improved by using the most recent convolution neural network architecture. The goal of our project is to close communication barriers with people who have speech and hearing impairments.

3.4 Problem Solution fit



Problem-Solution fit

TEAM ID: PNT2022TMID04184

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Deaf and dumb people are referred to as "Specially-abled" individuals.

6. CUSTOMER CONSTRAINTS CC

People with disabilities have trouble communicating with others. They become wary of unfamiliar surroundings and people as a result.

5. AVAILABLE SOLUTIONS AS

People with disabilities have trouble communicating with others. They become wary of unfamiliar surroundings and people as a result.

Explore AS, differentiate

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

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

Audio and text messages made using

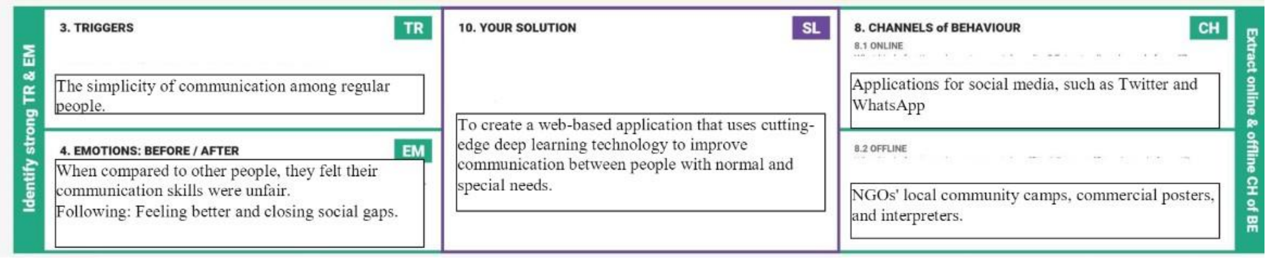
9. PROBLEM ROOT CAUSE RC

Since most people don't make an effort to learn sign language, communicating with people who have special needs can be challenging.

7. BEHAVIOUR BE

They look for mobile apps and interpreters to improve communication with everyday people.

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



Identify strong TR & EM

3. TRIGGERS TR

The simplicity of communication among regular people.

4. EMOTIONS: BEFORE / AFTER EM

When compared to other people, they felt their communication skills were unfair.
Following: Feeling better and closing social gaps.

10. YOUR SOLUTION SL

To create a web-based application that uses cutting-edge deep learning technology to improve communication between people with normal and special needs.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Applications for social media, such as Twitter and WhatsApp

8.2 OFFLINE

NGOs' local community camps, commercial posters, and interpreters.

Extract online & offline CH of BE

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
FR-3	User Interface	User friendly and can be handled easily.
FR-4	Collection of datasets	Information is gathered regarding the various signs and the corresponding voice.

4.2 Non-functional Requirements:

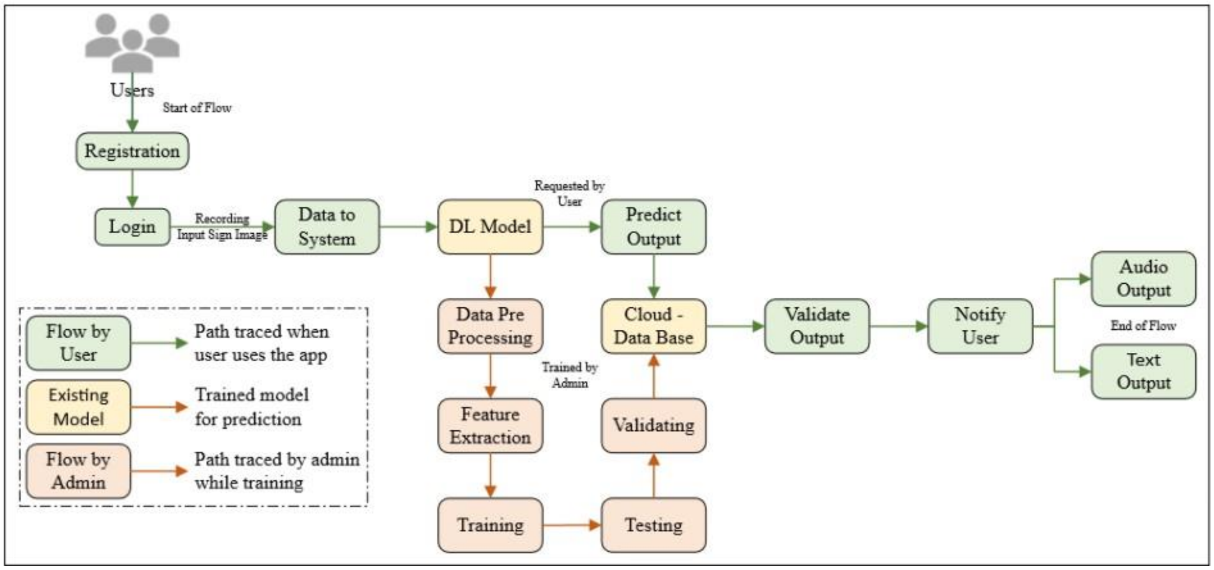
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Used to allow people with special needs to speak openly with people without any obstacles.
NFR-2	**Security**	The user of this website, their information, and their personal data are secured.
NFR-3	**Reliability**	After correctly determining the exact sign conversion for the given sign, their corresponding voice is then outputted with high accuracy.
NFR-4	**Performance**	The proposed approach is more effective and can be accessed by manyindividuals because it will be made available online and has an easy-to-use interface.
NFR-5	**Availability**	The website needs to be responsive in order to be usable on all types of devices, including computers, tablets, and mobile phones.
NFR-6	**Scalability**	The proposed model can be scaled up to become a more complex system that can recognise several sign languages and also translate into multiple conventional languages with enough cash and labor.

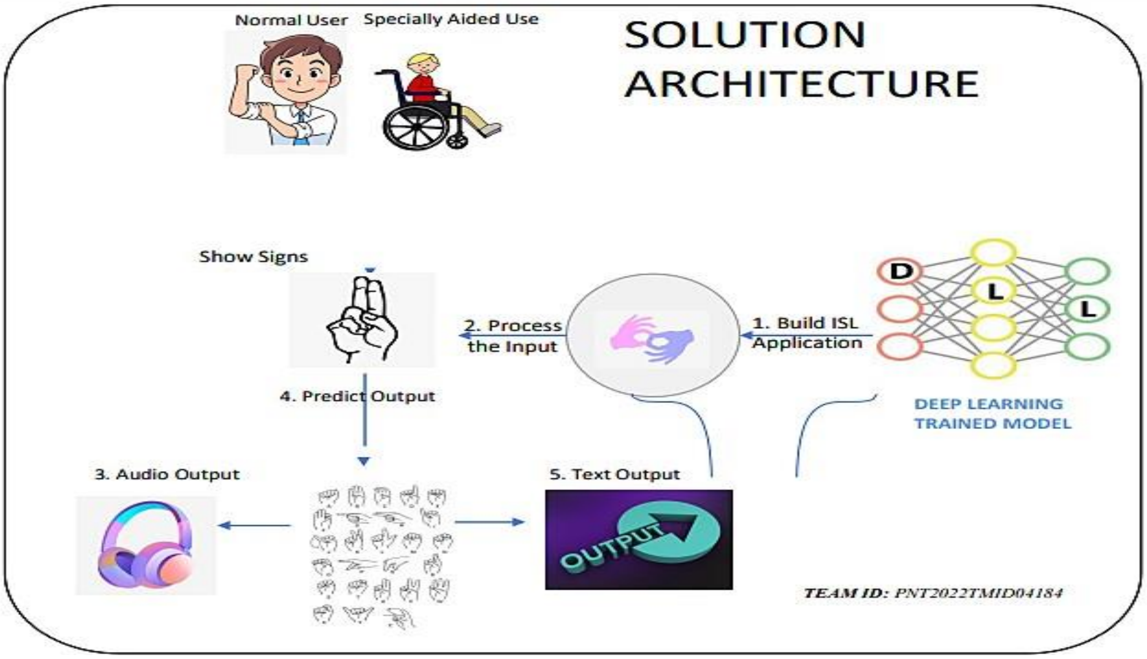
5 PROJECT DESIGN

5.1 Data Flow Diagram:

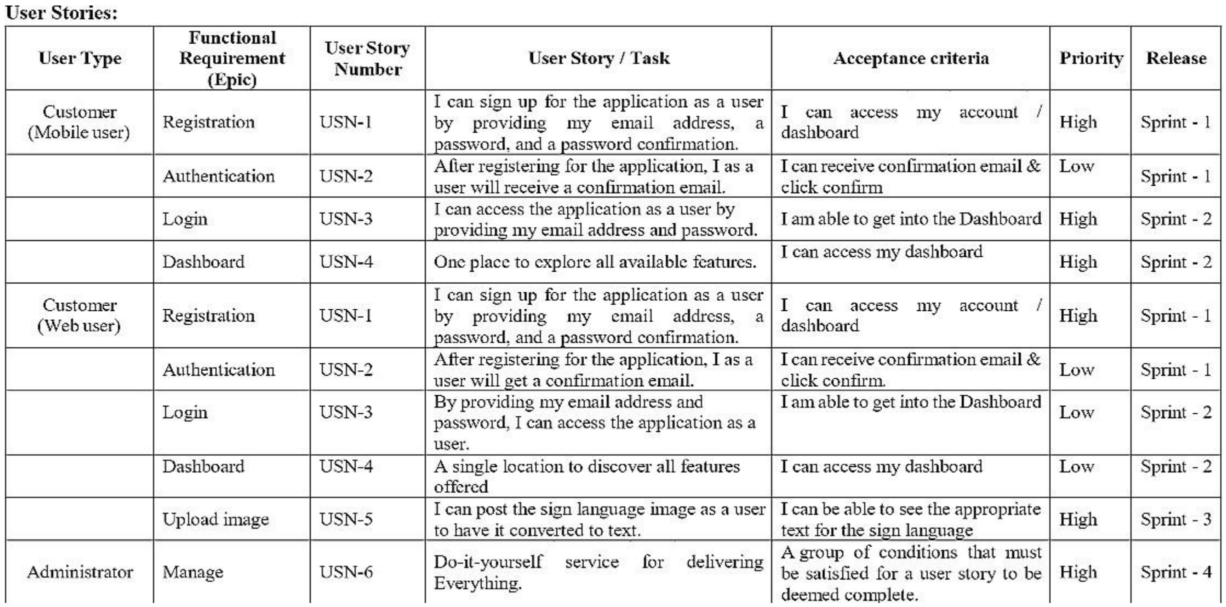
Data Flow Diagram:



5.2 Solution & Technical Architecture:



5.3 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	I can sign up for the application as a user by providing my email address, a password, and a password confirmation.	I can access my account / dashboard	High	Sprint - 1
	Authentication	USN-2	After registering for the application, I as a user will receive a confirmation email.	I can receive confirmation email & click confirm	Low	Sprint - 1
	Login	USN-3	I can access the application as a user by providing my email address and password.	I am able to get into the Dashboard	High	Sprint - 2
	Dashboard	USN-4	One place to explore all available features.	I can access my dashboard	High	Sprint - 2
Customer (Web user)	Registration	USN-1	I can sign up for the application as a user by providing my email address, a password, and a password confirmation.	I can access my account / dashboard	High	Sprint - 1
	Authentication	USN-2	After registering for the application, I as a user will get a confirmation email.	I can receive confirmation email & click confirm.	Low	Sprint - 1
	Login	USN-3	By providing my email address and password, I can access the application as a user.	I am able to get into the Dashboard	Low	Sprint - 2
	Dashboard	USN-4	A single location to discover all features offered	I can access my dashboard	Low	Sprint - 2
	Upload image	USN-5	I can post the sign language image as a user to have it converted to text.	I can be able to see the appropriate text for the sign language	High	Sprint - 3
Administrator	Manage	USN-6	Do-it-yourself service for delivering Everything.	A group of conditions that must be satisfied for a user story to be deemed complete.	High	Sprint - 4

6 PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning an

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect Dataset .	9	High	DWARAKNATH K, SWATHI K, HARINI K, DINESH KUMAR S, RAKSHANA S
Sprint-1		USN-2	Image preprocessing	8	Medium	DWARAKNATH K, SWATHI K, HARINI K, DINESH KUMAR S, RAKSHANA S
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	10	High	DWARAKNATH K, SWATHI K, HARINI K, DINESH KUMAR S, RAKSHANA S
Sprint-2		USN-4	Training the image classification	7	Medium	DWARAKNATH K, SWATHI K, HARINI K, DINESH KUMAR S, RAKSHANA S

			model using CNN			
Sprint-3	Training and Testing	USN-5	Training the model and testing the model’s performance	9	High	DWARAKNATH K, SWATHI K, HARINI K, DINESH KUMAR S, RAKSHANA S
Sprint-4	Implementation of the application	USN-6	Converting the input sign language images into English alphabets	8	Medium	DWARAKNATH K, SWATHI K

6.2 Sprint Delivery Schedule

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

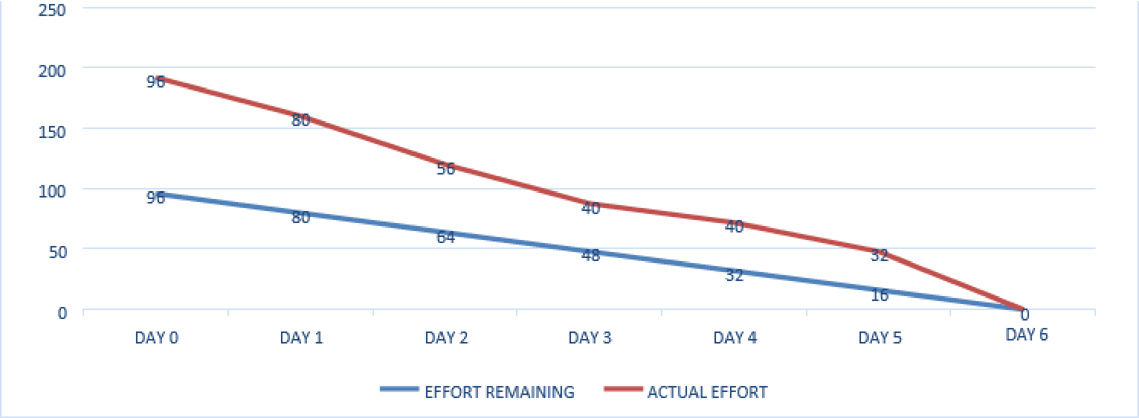
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	04 Nov 2022	5	04 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022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

Velocity:

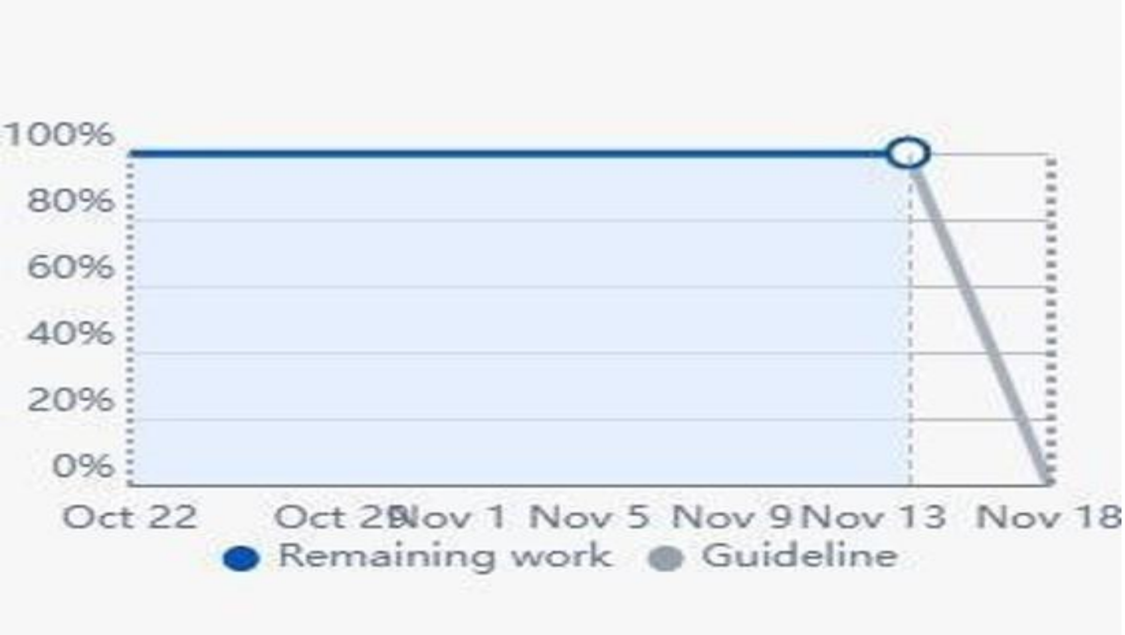
$$AV = \frac{\textbf{Sprint Duration}}{\textbf{Velocity}}$$

AV = 6/10 =0.6

Burndown chart:



SPRINT BURNDOWN CHART:

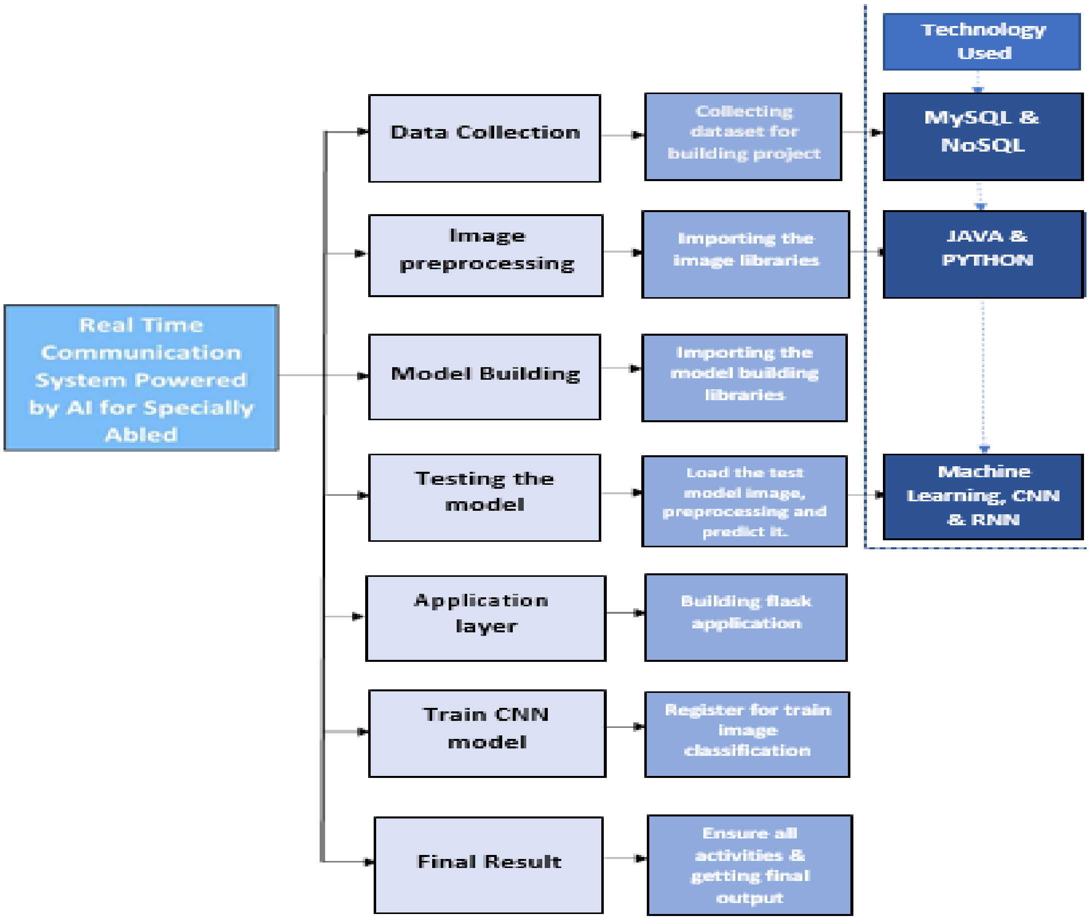


Milestone Activity Plan:

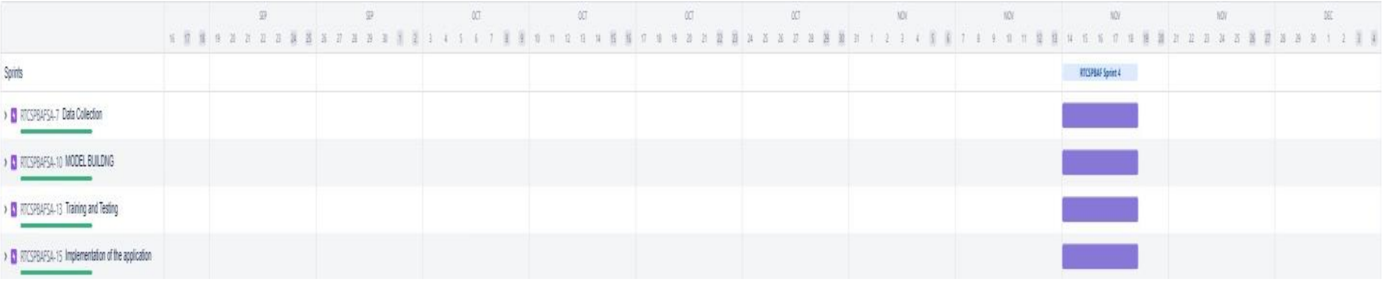
Milestone	Function (Epic)	Milestone Story Number	Story / Task
Milestone 1	Data collection	M1	In order to build our project, we are gathering data and creating two folders: onefor training and the other for testing.
Milestone2	Image preprocessing	M2	Importing libraries for image data generators and using theirfeatures to trainthe test set.
Milestone3	Model building	M3	Importing the libraries for model construction, setting up the model, Convolution layeraddition, layer pooling, layer flattening, layer dense layer addition Putting together the model and saving it.
Milestone4	Testing the model	M4	Initially import the packages. The test imageis then preprocessed and predicted whenthe model has been saved and loaded.

Milestone 5	Application layer	M5	Develop the HTML pages and the Flaskapplication.
Milestone6	Train CNNmodel	M6	Activate IBM Cloud and prepare theimage classification model.
Milestone7	Final result	M7	To ensure that all processes result in thedesired outcome.

MILESTONE ACTIVITY PLAN


Technology Used
MySQL & NoSQL
JAVA & PYTHON
Machine Learning, CNN & RNN
Real Time Communication System Powered by AI for Specially Abled
Data Collection
Collecting dataset for building project
Image preprocessing
Importing the image libraries
Model Building
Importing the model building libraries
Testing the model
Load the test model image, preprocessing and predict it.
Application layer
Building flask application
Train CNN model
Register for train image classification
Final Result
Ensure all activities & getting final output

6.3 Reports from JIRA



Sprints

RTCSPBAF Sprint 4

RTCSPBAFSA-7 Data Collection

RTCSPBAFSA-10 MODEL BUILDNG

RTCSPBAFSA-13 Training and Testing

RTCSPBAFSA-15 Implementation of the application

7 CODING & SOLUTIONING

(Explain the features added in the project along with code)

7.1 Libraries to be installed

```
pip install fer
pip install flask
pip install cv2
pip install numpy
pip install keras
pip install tensorflow
pip install  cvzone
pip install pyttsx3
pip install scikit-image
```

7.2 Real-Time Sign to Speech Translation

People who are unable to talk typically use sign language to communicate. The majority of people find it extremely challenging to communicate with silent persons since they are unable to grasp Universal Sign Language (unless they have studied it). The core of this project is a tool that allows silent people and others communicate with one another. Our technology makes use of a CNN-built model that can recognise sign languages in real time.

7.3 Real-Time Speech to Sign Conversion

We can use JavaScript to recognise voice thanks to the Web Speech API. JavaScript makes it very simple to recognise speech in a browser and then extract the text from the speech for use as user input. The voice is translated into text using the SpeechRecognition object, which is subsequently shown on the screen. This can be accomplished by our technology in real time. Any language that the user is trying to speak in can be recognised by it. But only the Chrome browser is supported for this API. The live example below may not function if you are viewing this example in another browser.

8 TESTING

8.1 Test Cases:

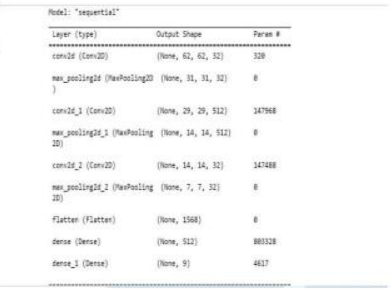
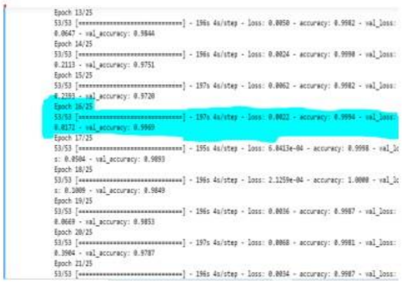
- Check to see that the options are visible when the user hits the URL.
- Check to see if the UI elements are displaying correctly.
- Check to see if the user can select any languages.
- Check to see if the user is being forwarded to the sign-to-speech website.
- Check to see if the program can convert the sign to speech.
- Check to make sure the user can leave the sign-to-speech page.

8.2 UAT Testing

Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	15	0	0	15
Security	2	0	0	2
Outsource Shipping	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8.3 Performance Testing

S.NO	Parameter	Values	Screenshot
1.	Model Summary	Total params: 1,103,721 Trainable params: 1,103,721 Non-trainable params: 0	
2.	Accuracy	Training Accuracy - 0.9994 Validation Accuracy -0.9969	

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        320

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 512)       147968

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 512)      0
 2D)

 conv2d_2 (Conv2D)           (None, 14, 14, 32)        147488

 max_pooling2d_2 (MaxPooling  (None, 7, 7, 32)         0
 2D)

 flatten (Flatten)           (None, 1568)              0

 dense (Dense)               (None, 512)               803328

 dense_1 (Dense)             (None, 9)                 4617

=================================================================
```

```
Epoch 13/25
53/53 [==============================] - 196s 4s/step - loss: 0.0050 - accuracy: 0.9982 - val_loss:
0.0647 - val_accuracy: 0.9844
Epoch 14/25
53/53 [==============================] - 196s 4s/step - loss: 0.0024 - accuracy: 0.9990 - val_loss:
0.2113 - val_accuracy: 0.9751
Epoch 15/25
53/53 [==============================] - 197s 4s/step - loss: 0.0062 - accuracy: 0.9982 - val_loss:
0.2393 - val_accuracy: 0.9720
Epoch 16/25
53/53 [==============================] - 197s 4s/step - loss: 0.0022 - accuracy: 0.9994 - val_loss:
0.0172 - val_accuracy: 0.9969
Epoch 17/25
53/53 [==============================] - 195s 4s/step - loss: 6.0413e-04 - accuracy: 0.9998 - val_lo
s: 0.0504 - val_accuracy: 0.9893
Epoch 18/25
53/53 [==============================] - 196s 4s/step - loss: 2.1259e-04 - accuracy: 1.0000 - val_lo
s: 0.1009 - val_accuracy: 0.9849
Epoch 19/25
53/53 [==============================] - 196s 4s/step - loss: 0.0036 - accuracy: 0.9987 - val_loss:
0.0669 - val_accuracy: 0.9853
Epoch 20/25
53/53 [==============================] - 197s 4s/step - loss: 0.0068 - accuracy: 0.9981 - val_loss:
0.3904 - val_accuracy: 0.9787
Epoch 21/25
53/53 [==============================] - 196s 4s/step - loss: 0.0034 - accuracy: 0.9987 - val_loss:
```

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

In [ ]:

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

In [ ]:

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

Out[ ]:

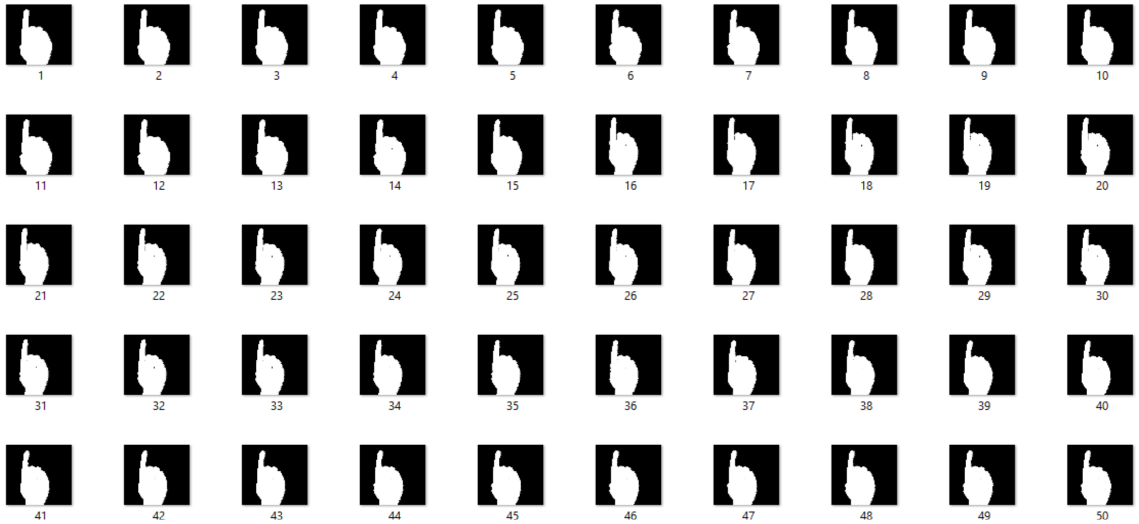
In [ ]:

```
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
```

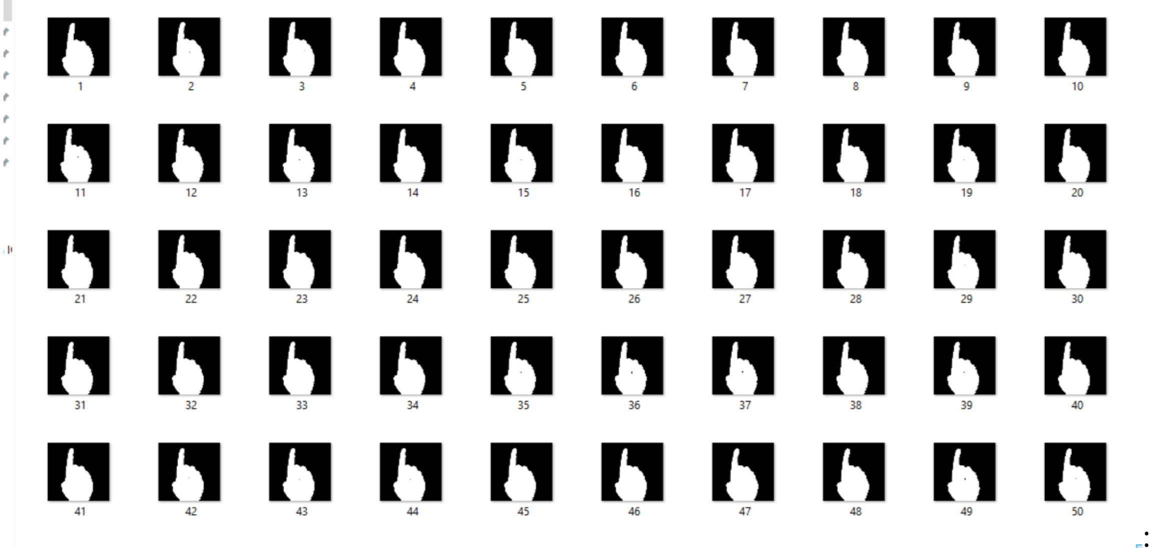
In [ ]:

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

TEST SET : (ALPHABET D)



TRAINING SET (ALPHABET D)



9 RESULTS

Performance Metrics

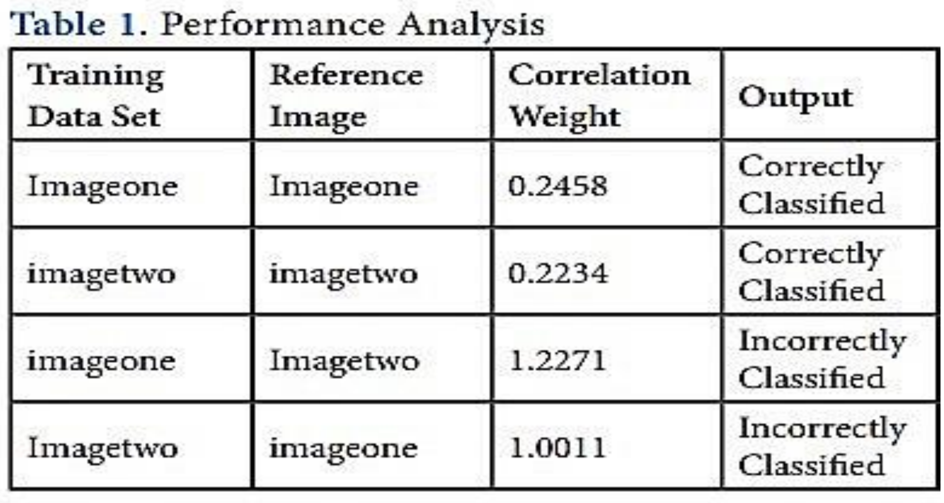
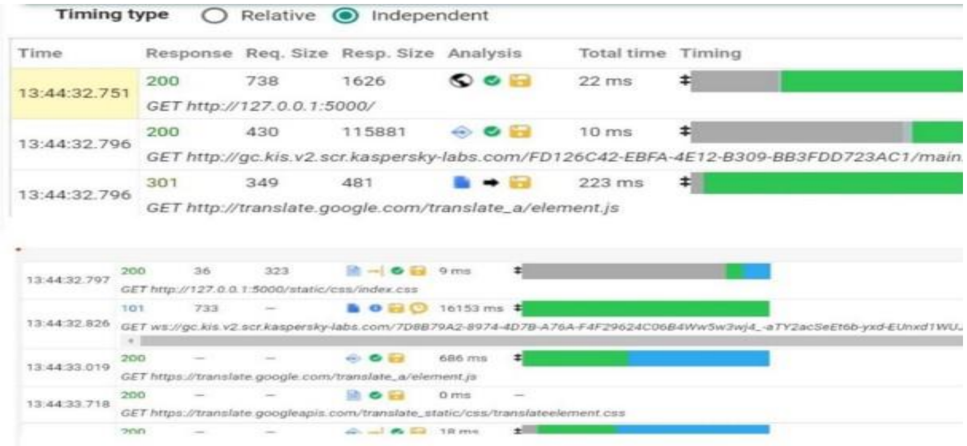


Table 1. Performance Analysis

Training Data Set	Reference Image	Correlation Weight	Output
Imageone	Imageone	0.2458	Correctly Classified
imagetwo	imagetwo	0.2234	Correctly Classified
imageone	Imagetwo	1.2271	Incorrectly Classified
Imagetwo	imageone	1.0011	Incorrectly Classified



Timing type ○ Relative ● Independent

Time	Response	Req. Size	Resp. Size	Analysis	Total time	Timing
13:44:32.751	200 GET http://127.0.0.1:5000/	738	1626		22 ms	
13:44:32.796	200 GET http://gc.kis.v2.scr.kaspersky-labs.com/FD126C42-EBFA-4E12-B309-BB3FDD723AC1/main.	430	115881		10 ms	
13:44:32.796	301 GET http://translate.google.com/translate_a/element.js	349	481		223 ms	
13:44:32.797	200 GET http://127.0.0.1:5000/static/css/index.css	36	323		9 ms	
13:44:32.826	101 GET ws://gc.kis.v2.scr.kaspersky-labs.com/7D8B79A2-8974-4D7B-A76A-F4F29624C06B4Ww5w3wj4_-aTY2acSeEt6b-yxd-EUnxd1WU.	733	—		16153 ms	
13:44:33.019	200 GET https://translate.google.com/translate_a/element.js	—	—		686 ms	
13:44:33.718	200 GET https://translate.googleapis.com/translate_static/css/translateelement.css	—	—		0 ms	—
	200	—	—		18 ms	

10 ADVANTAGES & DISADVANTAGES

ADVANTAGES

1. Sign language translation and detection in real time.
2. Developed model provides better accuracy.
3. Language is customizable.
4. User Interface is friendly.
5. There is real time speech to text conversion.
6. Data is secured and private.

DISADVANTAGES

1. The website would take some time to load.
2. Model still needs to be tested in a greater and macro level by wider range of data sets.
3. Customization of sign language is not available.
4. It is difficult to take notes while using the application.
5. The developed product works only in google chrome.
6. It is difficult to make calls using the developed application.

11 Conclusion:

With our method the frames are extracted from video. From the frames the hand gestures could be identified by using background subtraction techniques. The segmentation of the hand movements plays a vital role in such process. Sign languages, as spoken languages, have certain rules of grammar. These rules must be taken into account while transforming a sign language into a spoken language. In the end, adding a speech engine to speak the transformed text would help enhance ease of use. We are conducting further to develop a method for handling a large vocabulary. The detection capability of the system could be expanded to body gestures as well. The same technique could be used in a portable device.

12 Future Scope:

Researchers are working to improve hand gesture detection utilising the human nervous system. To recognise human gestures for automation, a system with a more sophisticated algorithm than those already presented and in use must be created.

13 Appendix

Source Code

Model Building

```
import cv2
import os os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
import numpy as np
from keras.models import Sequential
import matplotlib.pyplot as plt from keras.layers import Dense, Dropout,
Activation, Flatten
from keras.layers import Conv2D, MaxPool2D
from keras_preprocessing.image import ImageDataGenerator test_path =
'Dataset/test_set' train_path = 'Dataset/training_set'
train=ImageDataGenerator(rescale=1./255,zoom_range=0.2,shear_range=0.2,horiz
ontal_flip=T rue) test=ImageDataGenerator(rescale=1./255)
train_batches = train.flow_from_directory(directory=train_path,
target_size=(64,64), class_mode='categorical',
batch_size=300,shuffle=True,color_mode="grayscale")
test_batches = test.flow_from_directory(directory=test_path, target_size=(64,64),
class_mode='categorical', batch_size=300, shuffle=True,color_mode="grayscale")
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
input_shape=(64,64,1)))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Conv2D(512, (3, 3), padding="valid"))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Conv2D(32, (3, 3), padding="same"))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(512,activation ="relu"))
model.add(Dense(9,activation ="softmax"))
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
history = model.fit(train_batches,
batch_size=32,validation_data=test_batches,epochs=25) model.save('model.h5')
```

Model Testing

```
import keras from keras.models
import load_model
import cv2
import numpy as np
import os os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
val=['A','B','C','D','E','F','G','H','I']
model=load_model('model.h5')
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
                predict_x=model.predict(img)
                print(predict_x)
                predict=np.argmax(predict_x,axis=1)
                x=predict[0] print(val[x])
frame=cv2.imread(r"C:\Users\Akshaya\PycharmProjects\Realtime_Communicati
on_System_For_Specially_Abled\Dataset\test_set\B\1.png")
data=detect(frame)
```

Camera.py

```
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):
        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 Start
```

Main.py

```
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()
```

HTML File:

```
<!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>SmartBridge_WebApp_VideoTemplate</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="assets/css/Banner-Heading-Image.css">
    <link rel="stylesheet" href="assets/css/Navbar-Centered-Brand.css">
    <link rel="stylesheet" href="assets/css/styles.css">
</head>

<body style="background: rgb(39,43,48);">
    <nav class="navbar navbar-light navbar-expand-md py-3" style="background:
#212529;">
        <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><span style="color:
rgb(255,255,255);">Real-Time Communication
                    System Powered By AI&nbsp;For Specially Abled</span></a>
            <div></div>
        </div>
    </nav>
    <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: 640px;height: 480px;margin: 10px;min-height: 480px     in-
width: 640px;border-radius: 10px;border: 4px dashed rgb(255,255,255) ;">
                <img src="{{ url_for('video_feed') }}" style="width: 100%;height:
100%;color: rgb(255,255,255);text-align: center;font-size: 20px;"
                    alt="Camera Access Not Provided!">
            </div>
        </div>
        <div class="d-flex flex-column justify-content-center align-items-center"
style="margin-bottom: 10px;"><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="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="background: rgb(39,43,48);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">Artificial Intelligence has made it possible to
handle our daily activities
                                in new and simpler ways. With the ability to automate tasks
that normally require human
                                intelligence, such as speech and voice recognition, visual
perception, predictive text
                                functionality, decision-making, and a variety of other tasks, AI
can assist people with
                                disabilities by significantly improving their ability to get
around and participate in
                                daily activities.<br><br>Currently, Sign Recognition is
available <strong>only for
                                    alphabets A-I</strong> and not for J-Z, since J-Z alphabets
also require Gesture
                                Recognition for them to be able to be predicted correctly to a
certain degree of
                                accuracy.</p>
                        </div>
                    </div>
                </div>
                <div class="accordion-item" style="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="background: rgb(39,43,48);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 at VIT-Bhopal University during
SmartBridge AI Externship
                                Program.<br><br>1. <strong>Nirlov Deb</strong>
19BCG10067<br>2.
                                <strong>Kushagra</strong> 19BCG10025<br>3.
<strong>Kartik Dhasmana</strong> 19BCG10002
                            </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="{{ url_for('static',
filename='img/ASL_Alphabets.png') }}" width="100%"></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>
```

```
cv2.destroyAllWindows()
```

```
                    cv2.imwrite('image.jpg',copy)
                    copy_img = image.load_img('image.jpg', target_size=(64,64))
                    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)
                    return jpg.tobytes()
```

CSS File:

Navbar Centered Brand.Css

```
.bs-icon {
 --bs-icon-size: .75rem;
 display: flex;
 flex-shrink: 0;
 justify-content: center;
 align-items: center;
 font-size:  var(--bs-icon-size);
 width: calc(var(--bs-icon-size) * 2);
 height: calc(var(--bs-icon-size) * 2);
 color: var(--bs-primary);
}

.bs-icon-xs {
  --bs-icon-size: 1rem;
  width: calc(var(--bs-icon-size) * 1.5);
  height: calc(var(--bs-icon-size) * 1.5);
}

.bs-icon-sm {
  --bs-icon-size: 1rem;
}

.bs-icon-md {
  --bs-icon-size: 1.5rem;
}

.bs-icon-lg {
  --bs-icon-size: 2rem;
}

.bs-icon-xl {
  --bs-icon-size: 2.5rem;
}

.bs-icon.bs-icon-primary  {
  color: var(--bs-white);
  background: var(--bs-primary);
}

.bs-icon.bs-icon-primary-light {
  color: var(--bs-primary);
  background: rgba(var(--bs-primary-rgb), .2);
}

.bs-icon.bs-icon-semi-white {
  color: var(--bs-primary);
  background: rgba(255, 255, 255, .5);
}

.bs-icon.bs-icon-rounded {
  border-radius: .5rem;
}

.bs-icon.bs-icon-circle {
  border-radius: 50%;
}
```

Output 13.2

Github and Demo Link:

https://github.com/IBM-EPBL/IBM-Project-17195-1659630234