19L039 - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

JAYASURYAA G R (19L118)

SHRI PRIYADHARSHINI P (19L133)

VIBOOSITHASRI N S (19L147)

SOWMIYA S (20L417)

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

of Anna University

NOVEMBER 2022

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

Bonafide record of work done by

JAYASURYAA G R (19L118)

SHRI PRIYADHARSHINI P (19L133)

VIBOOSITHASRI N S (19L147)

SOWMIYA S (20L417)

Project report submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

of Anna University

NOVEMBER 2022

CONTENTS

ABSTRACT

Artificial intelligence is not designed to replace humans but rather to enhance their lives by helping do things that are difficult. This technology can be used for people with disabilities for additional support and guidance and make their life easier and simpler. With the ability to automate things, the project aims to develop an AI model that converts sign language into speech that can be understood by normal people. The human hand has remained a popular choice to convey information when speech cannot. With more features added, we can implement the most used technology, computer vision and neural networks and produce the respective outputs.

CHAPTER 1

INTRODUCTION

Communication between a 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 in hand sign language. In emergency times conveying their message is very difficult.

1.1. PROJECT OVERVIEW

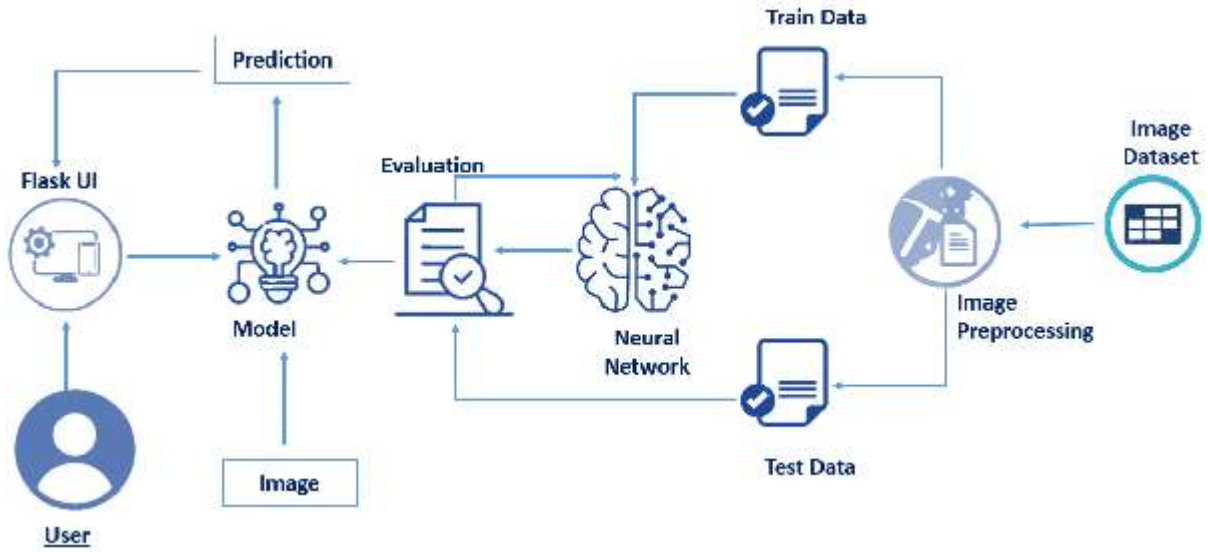


Figure 1.1. Technical Architecture

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.

1.2. PURPOSE

The main aim is to develop a system that converts 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.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Communicating with others is a big challenge for mute and blind people. The same holds true for staying connected to others in a world that's more and more digitized with the growing importance of social media and our dependence on the Internet. But technology and AI leave no one behind and can be at the service of people with disabilities. Five papers have been studied and the limitations have been analyzed. Details of five papers have been given in the next section.

2.2 REFERENCES

Paper 1: Real-time Two Hand Gesture Recognition with Condensation and Hidden Markov Models

A method for recognizing two-hand gestures is presented in this paper. The two basic methods used here for identifying hands are hand tracking and gesture recognition. This study provides a system that can identify 8 movements from real-time stereo color images. A Condensation Algorithm (Conditional Density Propagation Algorithm) is used to monitor the motion of the hands. The condensation function algorithm is resistant to computational load and capable of tracking objects with multiple locations. The performance of the system is significantly impacted by choosing effective features to recognize the hand gesture path. Three fundamental characteristics are position, orientation, and velocity. The system performs a classification as its last step. Additionally, a full training for the initialized HMM parameters by the discrete vector is performed using the Baum-Welch algorithm to create a gesture database. The proposed algorithm for distinguishing hands, localization of the hands and classification of the gestures gives good results.

Paper 2: Gesture Recognition System

In this paper, they focus on posture and gesture recognition. An algorithm using Singular value Decomposition - Principal Component Analysis (SVD- PCA) and a feed-forward artificial neural network is designed. Posture Recognition features are extracted using the SVD-PCA approach and used for training and classifying. Data is prepared that shows an image containing the hand and upper body portion and through this image skin pixels and edges are detected which are used in gesture-posture recognition. Skin pixels are detected using RGB color space and then some specific morphological operations are performed. Gray conversion is done on the input image and then a canny edge detection algorithm is used for the detection of edges and contours are obtained. The proposed method tracks posture in the upper body only. It can be extended to complete body posture recognition. But here, in the proposed technique, the background is uniform and only some types of postures are recognized.

Paper 3: Recognition Based on Hand Postures and Trajectories by Using Dataglove: A Fuzzy Probability Approach

Posture detection for sign language is mainly done in 2 ways:

1. Using image processing algorithms
2. Using embedded IoT based gloves

This paper has focused on 2nd method. A data glove has been designed which is capable of recording hand movements, both the position of the hand and its orientation as well as finger movements; it is capable of simple gesture recognition and general tracking of three-dimensional hand orientation. Hidden Markov Models (HMM) are an ideal approach for hand gesture recognition. HMM is a statistical model where the distributed initial points work well and the output distributions are automatically learned by the training process. American Sign Language (ASL) is the language most commonly used by the American deaf community. ASL uses hand, body, and facial expressions to make signs and convey ideas or feelings. At the end, adding a speech engine to speak the translated text would help enhance ease of use.

Paper 4: Hand Gesture Recognition Using Deep Learning

Gesture recognition - mathematical interpretation of a human motion by a computing device. This application belongs to the domain of hand gesture recognition which is generally divided into two categories i.e., contact-based, and vision-based approaches. For hand shape recognition, the classifier transfer learning over a pre trained CNN that is initially trained on a large dataset work, VGG16 a CNN architecture 13 convolution layers followed by 3 fully connected layers. recognize eleven hand shapes, hence CNN is trained as a classifier using transfer learning methods. 55 thousand self-created image datasets out of which 70 percent were used for training and rest for testing. Unidirectional hand gestures require shape and direction of motion of hand for commanding whereas multidirectional gestures require the position of hand along with its shape. Out of five dynamic hand shapes three are used for unidirectional gestures namely: swap, scroll and zoom, and remaining are used for multidirectional gestures of pointer and cursor.

Paper 5: An Efficient Approach for the Recognition of Hand Gestures from Very Low-Resolution Images

The recognition of hand gestures has become an area of active research in the field of computer vision and machine learning. The focus of the proposed method is the development of a simple, robust, and effective method for the recognition of hand gestures. The Information extracted is based on the geometrical structure of the hand and not on colour, size, or brightness. The low-resolution image containing the hand gesture needs to be pre-processed before the generation of the mask for the gesture. This involves the conversion of the RGB image to grayscale. The proposed method recognizes the hand gesture from very low resolution images. The method is made robust to both the lighting conditions as well as the orientation of the hand gesture by taking into consideration the geometrical structure of hand. Hand gestures recognized by this method are the number of fingers raised by the person. This type of systems can be efficiently used for controlling various application like home automation by capturing image from a very far distance such as a corner of the room.

2.3 PROBLEM STATEMENT DEFINITION

People with disabilities have limited access to employment opportunities, and even when they do, they must deal with a lack of understanding about their potential at work. Only those with special needs are taught sign language, and because the average person has no idea how it works, there is a communication gap. Specially abled people find it even harder to get assistance in an emergency. They may also need special assistance to navigate non-emergency normal environments. It has always been difficult to communicate with someone who is deaf-mute. It is very challenging for mute people to communicate with non-mute people. Creating a system that will be a fit solution for this problem will ease their way of living.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Suresh, one with hearing and speech disabilities	have a system or a device that replace my disabilities	could not find such a system	such a system is rarely available and does not effectively solves my problem	clueless
PS-2	a deaf-mute	find a system that replaces my disabilities so that I can overcome my inferiority complex and mingle with others	could not find one	no technology application is effective and user friendly	clueless
PS-3	an illiterate deaf-mute	have a system that helps me understand others speech as well as convey my message to normal people	such system which can solve my problem is not available	they convert speech to text and not as sign-language, which I cannot read	annoyed
PS-4	a person with hearing and speech disabilities	use a mobile application that facilitates verbal communication even during emergency situations with people who do not know sign language	no device in the market satisfies me	they are standalone devices and cannot be carried always	helpless
PS-5	a deaf-mute	live independently without others help to communicate	no product in the market is able to replace my sign-language translator	they are not as effective as human	frustrated

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

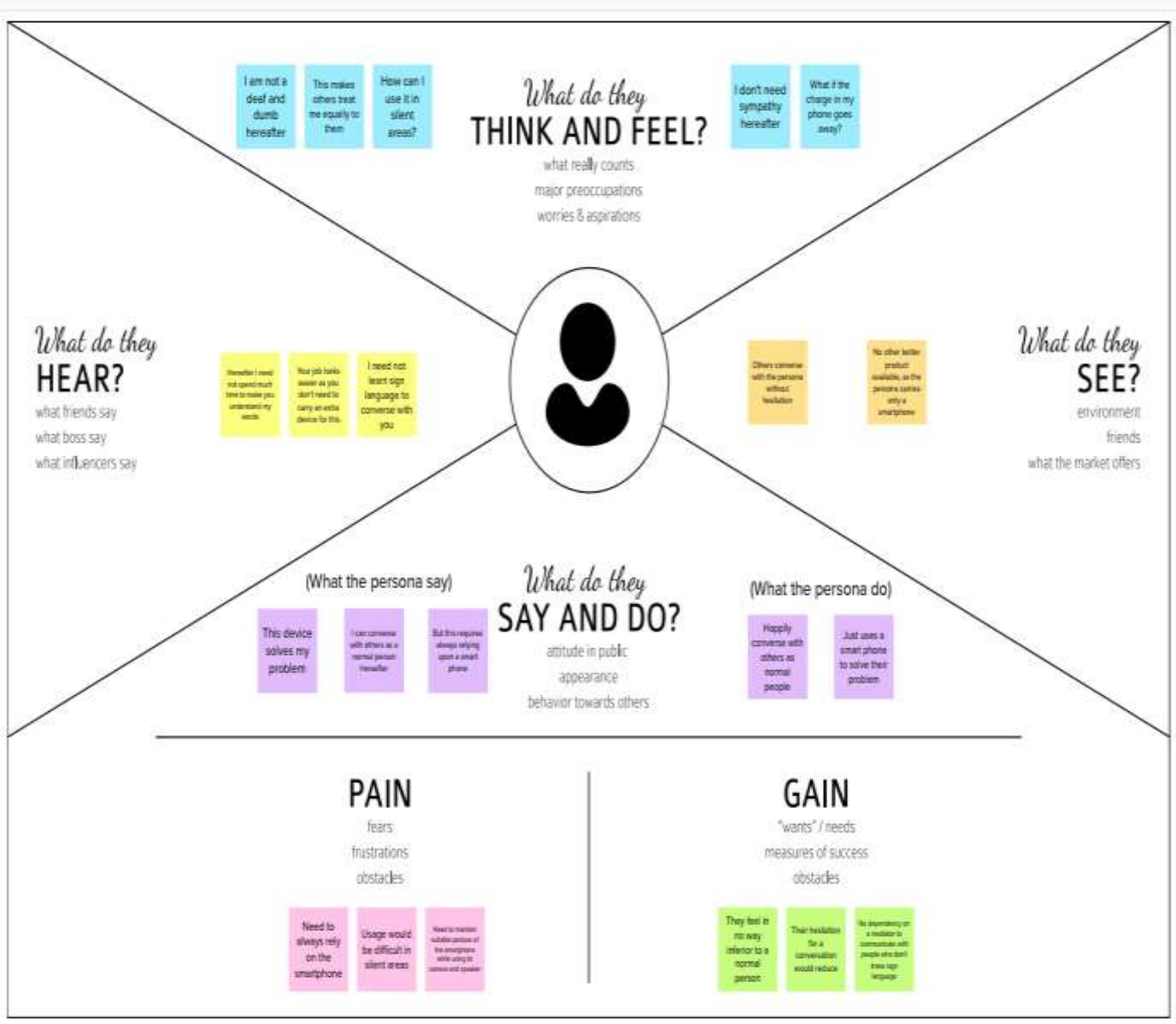
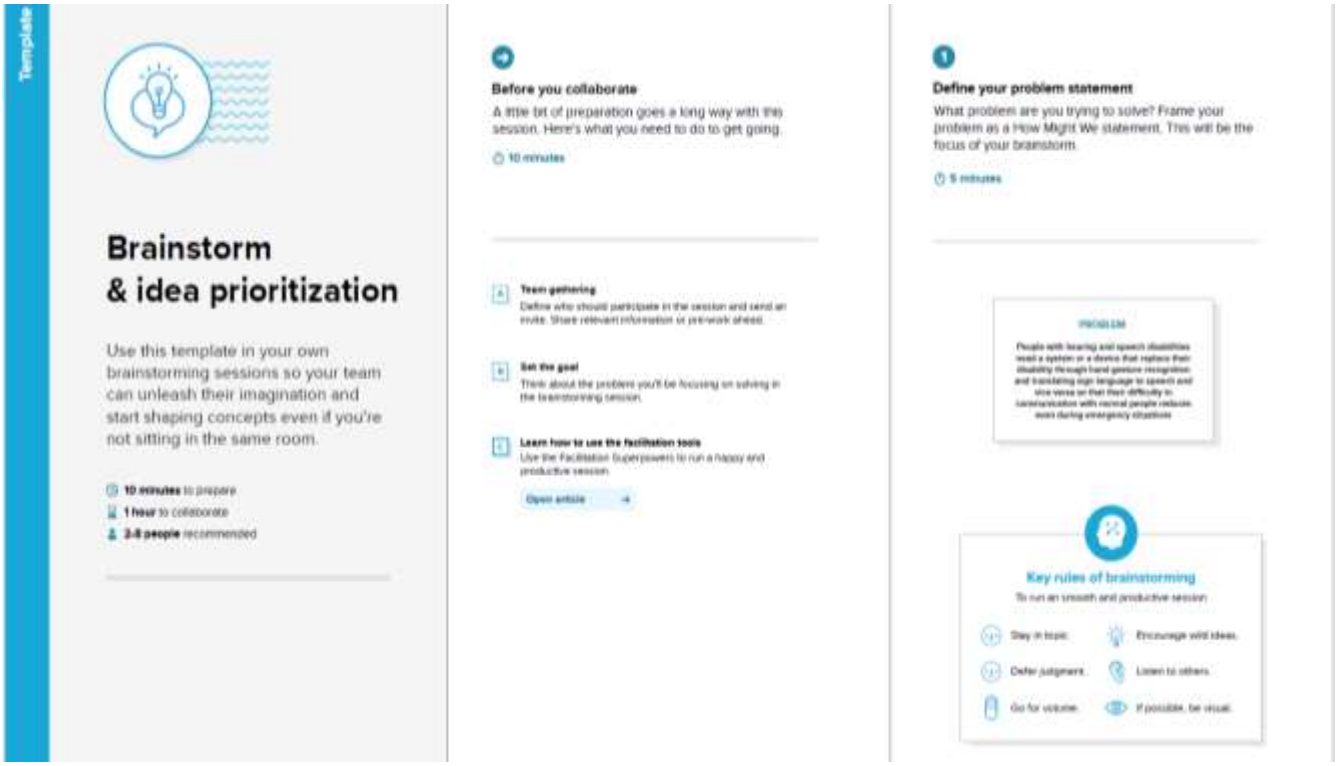


Figure 3.1. Empathy Map

3.2 IDEATION AND BRAINSTORMING

STEP 1 - TEAM GATHERING, COLLABORATION, AND SELECTING THE PROBLEM STATEMENT



STEP 2 - BRAINSTORMING, IDEA LISTING, AND GROUPING

Shri Priyadharshini P

Along with sign language detection, will facial expression also matter in a conversation? Can that be taken into account for our application?

Instead of maintaining suitable posture of the smart phone for camera and speaker usage, making a small dedicated product for this might solve the purpose in simpler way

Possibility of integrating it with Google Assistant, Alexa, Siri, etc will enhance its usage

Along with different language for speech output, applying the same for speech to text conversion will make it duplexer

Sowmiya S

Since there are different kinds of sign language, conversion from one sign language to another can break the barrier

We can facilitate conversion of sign language to emojis for fun remote communications

Converting sign language and giving option for either displaying it as text or speech or both, will facilitate communication even in silent areas

We can integrate the app with calendars, to-do list, etc. and give reminders in sign language

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Based on automatic, easy and quick response...

We can integrate a chat-bot so that it automatically responses to commonly used sentences.

Possibility of integrating it with Google Assistant, Alexa, Siri, etc will enhance its usage

We can facilitate conversion of sign language to emojis for fun remote communications

We can have daily used sentence or responses as one-texts so that each time they do not have to repeat them. This reduces the time gap between actions, conversion and conveying.

Based on increasing App features and User Experience

Providing recording feature for the final speech output will help them sharing thoughts to remote people

We can facilitate sign language conversion of voice mails/messages which can make them use social media and other apps as normal people

Parental safety monitoring kind of system can be deployed for the child users which can protect them from abusers

Audios and sign language actions can be segregated and saved in clouds. This will allow easy access.

We can integrate the app with calendars, to-do list, etc. and give reminders in sign language

Converting sign language and giving option for either displaying it as text or speech or both, will facilitate communication even in silent areas

Based on marketing and globalizing the product

Using as many languages as possible for speech output will globalize the product

An API/extension can be made which can be attached with online meet platforms which can detect sign languages through their video and convert it to captions for other people

Along with different language for speech output, applying the same for speech to text conversion will make it duplexer

Since there are different kinds of sign language, conversion from one sign language to another can break the barrier

Based on technical ideas and improvements in the product

Along with CNN, we can try other possibilities like Fuzzy Logic systems and analyze the results for better approach

As vision based technique is used, device has to be kept at some distance from the user. So gesture controller and voice controller can be added to control the device touchless.

Along with sign language detection, will facial expression also matter in a conversation? Can that be taken into account for our application?

We can either use image recognition algorithms (vision based) to decode the signs or like gloves that can be worn in hands attached with sensors, basically an wireless gloves.

A dedicated device for this conversion can be made (for eg, that uses RTOS for instant translation)

Instead of maintaining suitable posture of the smart phone for camera and speaker usage, making a small dedicated product for this might solve the purpose in simpler way

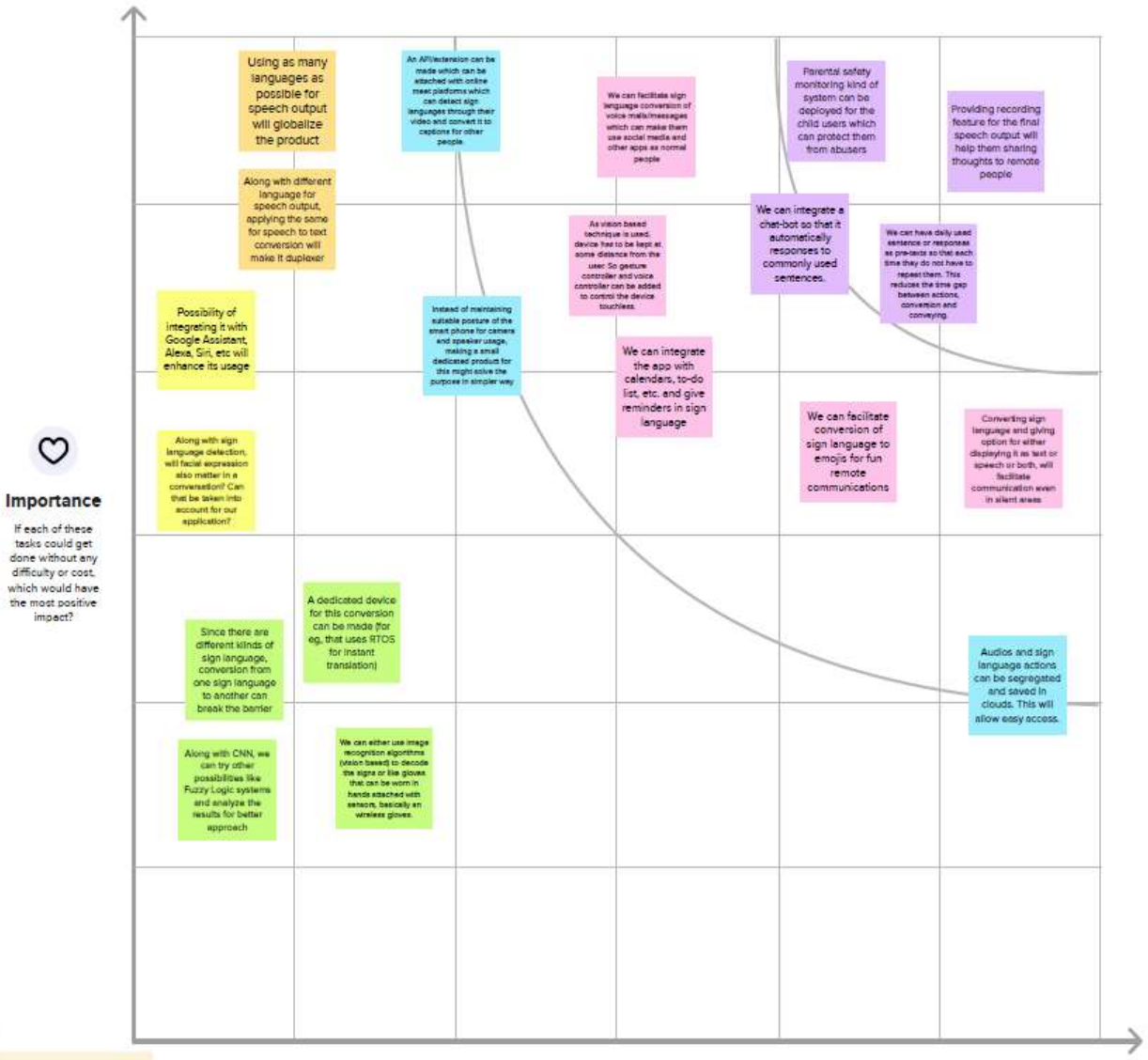
STEP 3 - IDEA PRIORITIZATION

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



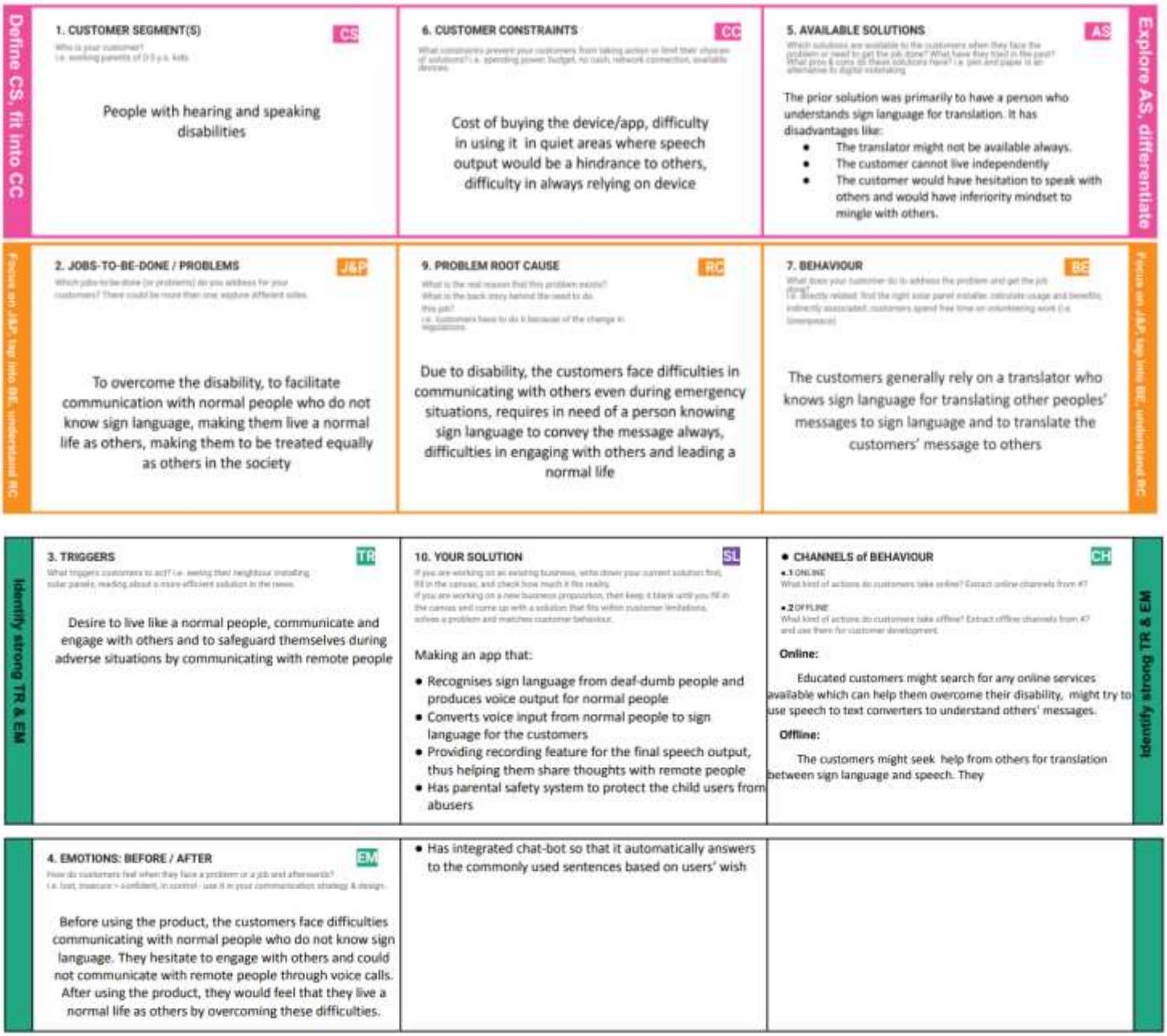
TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

3.3 PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	People who have hearing and speech disabilities find difficulties in communicating with others, especially with those who do not know sign language.
2	Idea / Solution description	The deaf-dumb people need a system or a device which can convert sign language to speech and text and can convert speech to sign language, so that they can overcome their disability.
3	Novelty / Uniqueness	Brainstorming and prioritising of ideas brought to the decision of having the following unique and novel features in it: 1. Providing recording feature for the final speech output will help them sharing thoughts to remote people. 2.We can integrate a chat-bot so that it automatically responds to commonly used sentences. 3.Parental safety monitoring kind of system can be deployed for the child users which can protect them from abusers.
4	Social Impact / Customer Satisfaction	1.The customer feel in no way inferior to a normal person. 2.Their hesitation for a conversation would reduce. 3.No dependency on a mediator to communicate with people who don't know sign language.
5	Business Model (Revenue Model)	1.Revenue can be generated by having advertisements in free space of app UI by avoiding protruding ads as it serves as a frequently used essential app for the customers. 2. Premium subscriptions can be introduced for educated and working customers which have the facility of speech conversion from text on the web, documents, mails, articles, etc.
6	Scalability of the Solution	The scalability of the product can be increased by: 1.We can integrate the app with calendars,to-do-list,etc. And give reminders in sign language. 2. Gesture controller and voice controller can be added to control the device touchless. 3.An API/extension can be made which can be attached with online meet platforms that can detect sign languages through their video and convert it to captions for other people.

3.4 PROBLEM SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

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 trusted third party authentication like google, microsoft, etc.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP (If third party authentication was not used)
FR-3	Account setup	This includes attaching required documents like valid ID proof, PwD certificate, etc
FR-3	Device Permission	Permission request for camera access - to get real time gestures and to show sign-language output Permission request for speaker access - to get real time speech input and to give translated speech output Permission for storage access - to facilitate the user to store important messages
FR-4	User input through app UI whether to start gesture recognition or to start speech recognition	Start listening to the speech or start capturing video based on user input.
FR-5	User input through app UI whether to stop the process	Stop listening/capturing and producing suitable output as video or gestures after recognition
FR-6	User input through app UI whether to start producing output	After processing the input, output should be produced as voice or gestures based on the input taken
FR-7	Displaying options for recording the output for future reference*	Start storing the result in the device if the user preferred to record it
FR-8	Displaying another option for sharing it to others	Starting to share the result to other apps based on user input
FR-9	Displaying option for start new recording or video capturing	Start listening to the speech or start capturing video based on user input.

FR-10	Providing option for editing user details and uploading PwD certificate in Account Settings page	Changing user details and storing the PwD certificate in the database based on user input.
FR-11	Providing option to change language (from the options available) of the app in Settings page	The UI will be changed to the user selected language
FR-12	Providing option for setting up child safety monitoring system in Settings Page*	Linking the apps in parent and child's devices. Providing option for the parent to view all the gestures and speeches from the child's device

4.2 NON-FUNCTIONAL REQUIREMENTS

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

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	IBM Watson services must be used to convert the output speech to users' desired language. The UI will contain only commonly used commands. It will be changed to users' desired language selected at the Settings page. These features make the app user friendly for various users, with even the illiterate users to easily adapt to the app.
NFR-2	Security	Registration of the users includes email/phone number confirmation. Trusted third party authentication will also be available. These features enhance the security. Also the child safety monitoring system protects the children from abusers
NFR-3	Reliability	Most of the features in the app work without the internet. Some features which require the internet like updating of details in the database, will be temporarily stored in the device when internet connectivity is not detected. It will start updating the database soon after the network is available. The quality of the AI model used will be periodically improved and app updates will be released. The performance of the implemented model will be collected based on users' preference.

NFR-4	Performance	Since the app works in real time, it should have instantaneous response. The latency in data processing after getting the input must be nearly zero.
NFR-5	Availability	The registration details of the user and other structural data must be stored in the database so that if the user changes the device or reinstall the app, he/she can continue without any difficulty. The AI model must be periodically developed and updates must be released. This new update availability must be notified to the users through the app. The user must be able to update during his/her desired time
NFR-6	Scalability	The app must be compatible to all the web browsers and to the Android version of at least 4.0. Most of the functionalities in the app must require no internet connection so that it can be used in remote areas. UI and the speech input/output must be changeable to users' desired language. These measures covers large users population.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

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

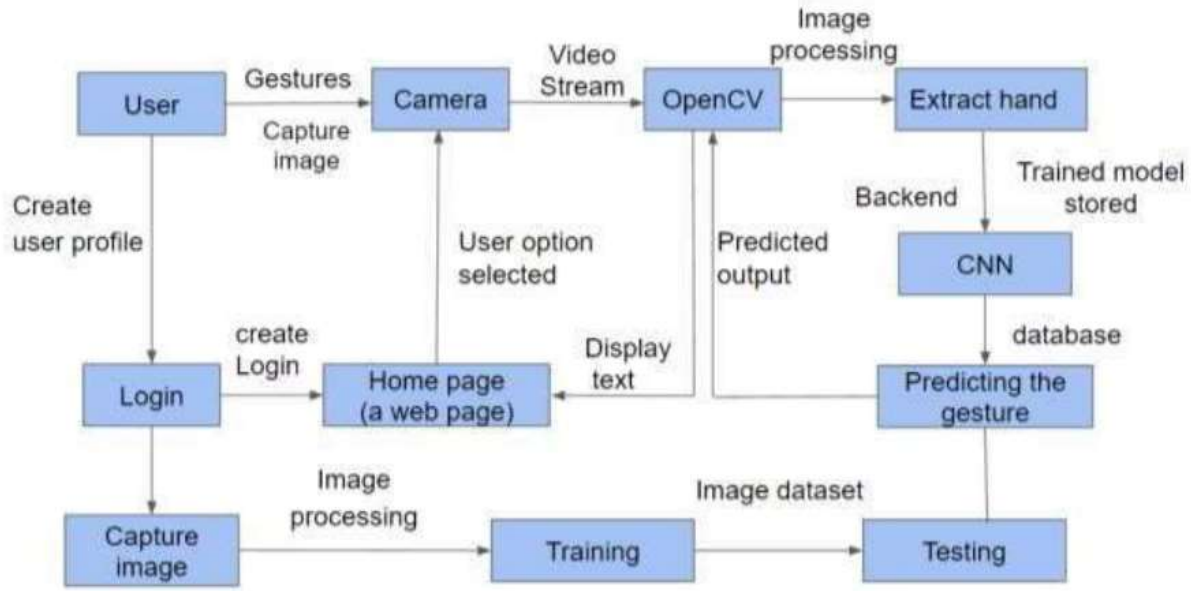


Figure 5.1. Data flow diagram

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

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.

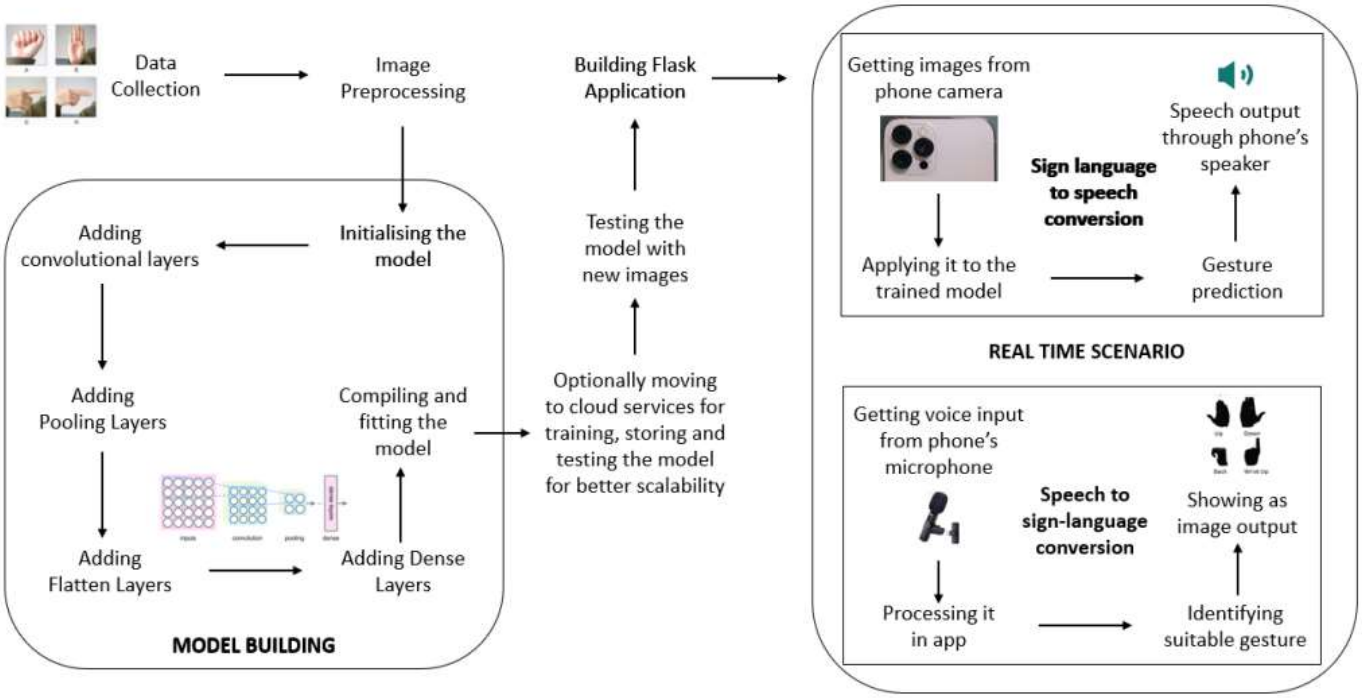


Figure 5.2 Solution Architecture of the project

TECHNICAL ARCHITECTURE

Technical Architecture includes the following:

1. All the processes (As an application logic /Technology Block)
2. Infrastructural demarcation (Local / Cloud)
3. External interfaces (third party API's etc.)
4. Data Storage components / services
5. Interface to machine learning models

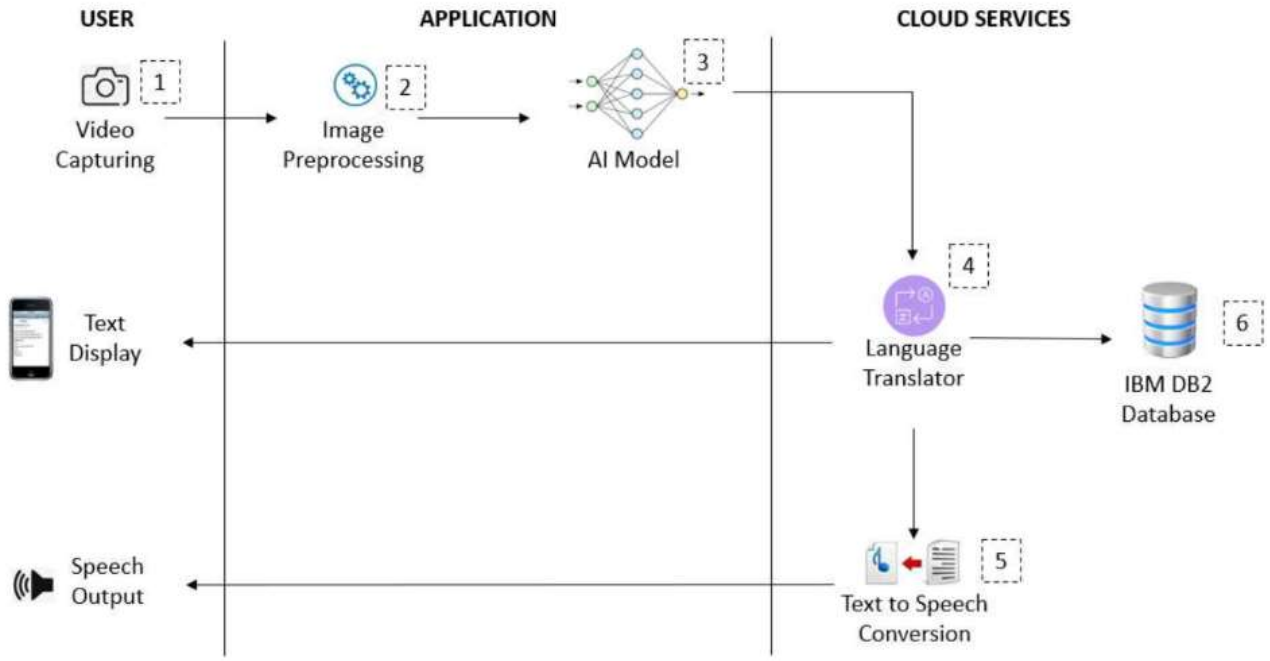


Figure 5.3 Technical Architecture of the project

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User story Number	User story/Task	Acceptance Criteria	Priority	Release
Customer (mute)	UI,AI Model	USN-1	As a user, who has trouble speaking, I want to convey my message through sign language to normal people for which I need a system that can convert sign-language to speech and text in their mother-tongue.	I can use button in the UI for converting my sign language to text and speech in desired language .	High	Sprint-1
Customer (deaf)	UI,AI Model	USN-2	As a user, who has trouble hearing, I want the text of the language spoken by others in my mother-tongue so that I can understand others message.	I can use the button in the UI for converting speech to text in my desired language.	High	Sprint-2
Customer (deaf and mute)	UI,AI Model	USN-3	As a user, who has trouble hearing and speaking, I want text of the language spoken by others in my mother-tongue as well as I want to convey my message through sign language to normal people in their mother tongue.	I can use button in the UI for converting others speech to text as well as for converting my sign language to text and speech in desired language.	High	Sprint-2

Customer (all application users)	Registration	USN-4	As a user, I can register for the application through my email and password.	I can create an account in the app with security	Medium	Sprint-3
		USN-5	As a user, I will receive confirmation email once I have registered in the application through email and password	I can trust that my account creation is authenticated	Medium	Sprint-3
	Login	USN-6	As a user, I can log into the application by entering email & password	I can log-in to my account in the app with security	Medium	Sprint-4
	3rd party APIs	USN-7	As a user, I can directly log into the application through trusted 3rd parties like Google, Microsoft, Facebook, etc.	I can log-in to my account very quickly and with high security	Low	Sprint-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User story Number	User story/Task	Story points	Priority	Team members
Sprint-1	UI,AI Model	USN-1	As a user, who has trouble speaking, I want to convey my message through sign-language to normal people for which I need a system that can convert sign-language to speech and text in their mother tongue	20	High	Jayasuryaa G R
Sprint-3	UI,AI Model	USN-2	As a user, who has trouble hearing, I want the text of the language spoken by others in my mother-tongue so that I can understand others message	15	High	Jayasuryaa G R
Sprint-3	UI,AI Model	USN-3	As a user, who has trouble hearing and speaking, I want text of the language spoken by others in my mother-tongue as well as I want to convey my message through sign-language to normal people in their mother-tongue	5	High	Shri Priyadharshini P
Sprint-3	Registration	USN-4	As a user, I can register for the application through my email and password	10	Medium	Shri Priyadharshini P
Sprint-3		USN-5	As a user, I will receive confirmation email once I have registered in the application through email and password	10	Medium	Sowmiya S

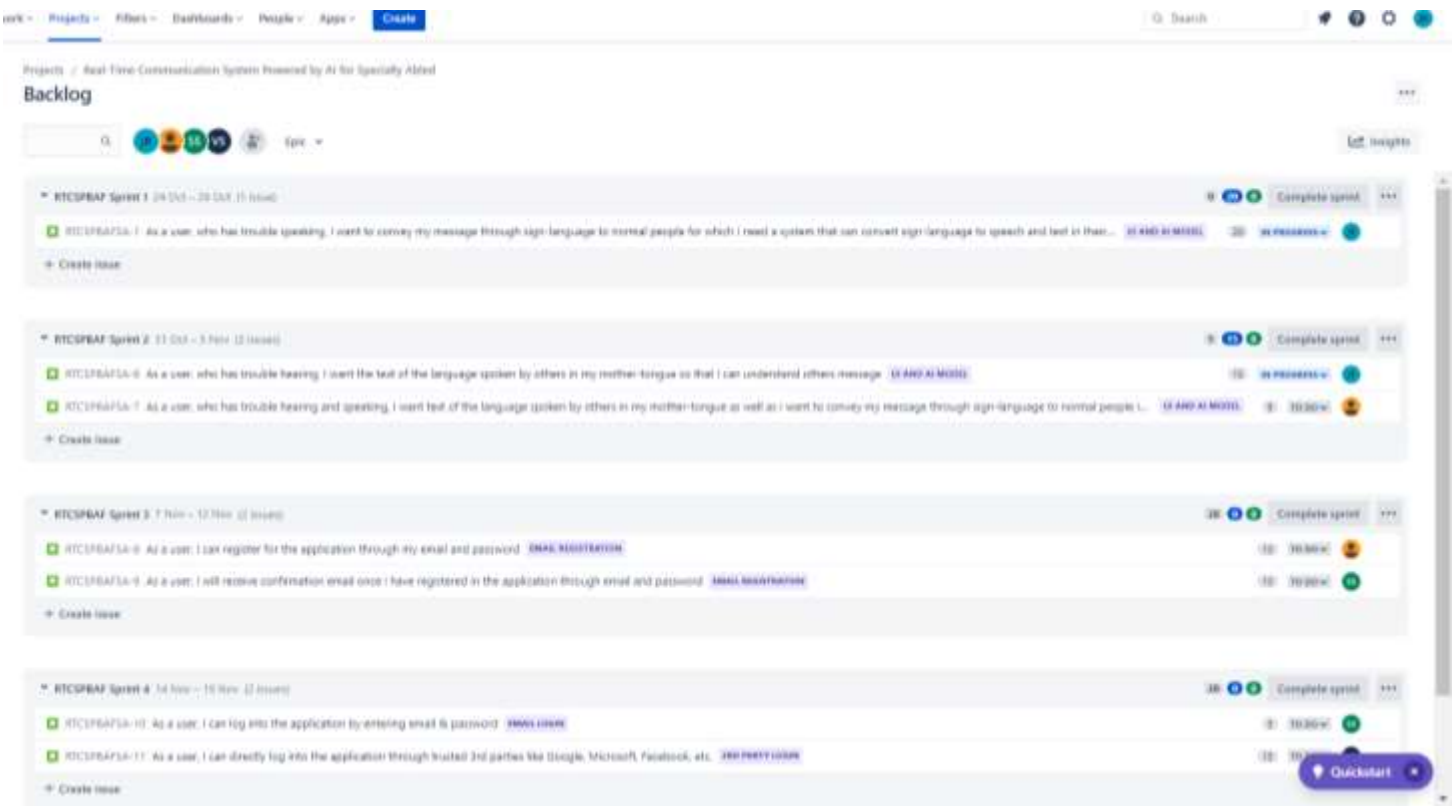
Sprint-4	Login	USN-6	As a user, I can log into the application by entering email & password	5	Medium	Sowmiya S
Sprint-4	3rd party APIs	USN-7	As a user, I can directly log into the application through trusted 3rd parties like Google, Microsoft, Facebook, etc.	15	Low	Viboosithasri N S

6.2 SPRINT DELIVERY SCHEDULE

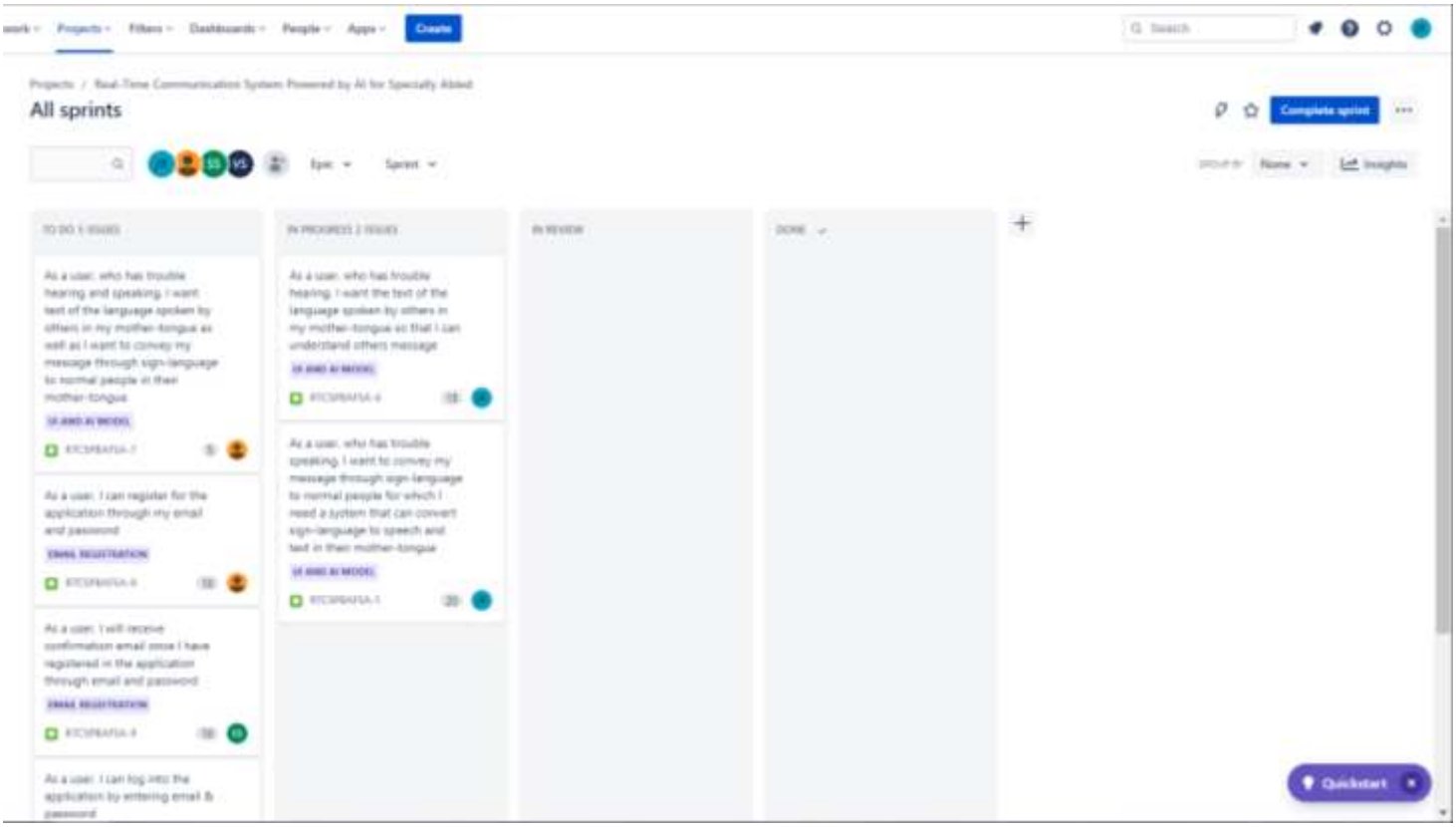
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA

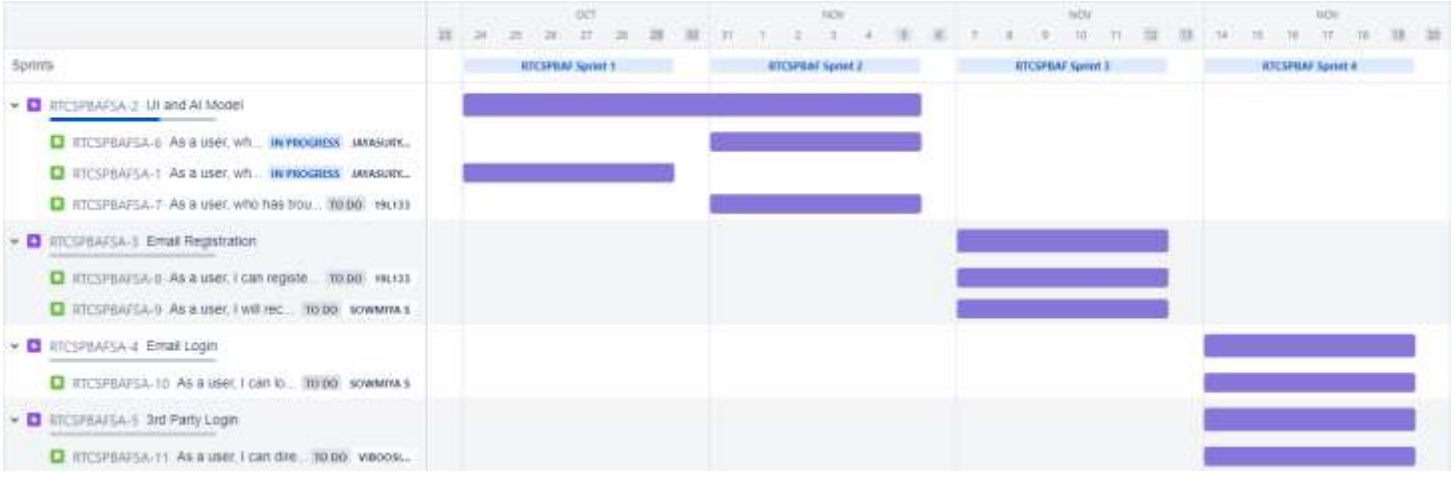
Backlog Page



Board Page



Road Map



CHAPTER 7
CODING AND SOLUTIONING

7.1 MODEL BUILDING CODE AND RESULTS

```
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers import BatchNormalization
from keras.callbacks import LearningRateScheduler, ReduceLROnPlateau
from keras.models import load_model
from keras.optimizers import Adam
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay
from skimage.transform import resize, rotate
import numpy as np
import cv2
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow


train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.1,
    zoom_range=0.1,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    validation_split=0.2,
    )
# test_datagen = ImageDataGenerator(rescale=1./255)


x_train = train_datagen.flow_from_directory(
    'gdrive/MyDrive/Dataset5',
    target_size=(28,28),
    batch_size=64,
    # save_to_dir='Dataset4/aug_set',
    class_mode='categorical',
    color_mode="grayscale",
    subset="training",
    )
x_test = train_datagen.flow_from_directory(
    'gdrive/MyDrive/Dataset5',
    target_size=(28,28),
    batch_size=64,
    class_mode='categorical',
    color_mode="grayscale",
    subset="validation",
```

```
    )
# x_test = train_datagen.flow_from_directory('Dataset/test_set', target_size=(64,64), batch_s
```

```
Found 22413 images belonging to 9 classes.
```

```
model = Sequential()
```

```
model.add(Convolution2D(75, (3,3) , strides = 1 , padding = 'same' , activation = 'relu' , in
model.add(BatchNormalization())
model.add(MaxPooling2D((2,2) , strides = 2 , padding = 'same'))
model.add(Convolution2D(50 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPooling2D((2,2) , strides = 2 , padding = 'same'))
model.add(Convolution2D(25 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2,2) , strides = 2 , padding = 'same'))
model.add(Flatten())
model.add(Dense(units = 512, activation = 'relu'))
model.add(Dropout(0.6))
model.add(Dense(units = 9 , activation = 'softmax'))
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 75)        750

 batch_normalization (BatchN  (None, 28, 28, 75)       300
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 14, 14, 75)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 50)        33800

 dropout (Dropout)           (None, 14, 14, 50)        0

 batch_normalization_1 (Batc  (None, 14, 14, 50)       200
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 50)         0
 2D)

 conv2d_2 (Conv2D)           (None, 7, 7, 25)          11275

 batch_normalization_2 (Batc  (None, 7, 7, 25)         100
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 25)         0
 2D)
```

```
 flatten (Flatten)           (None, 400)               0

 dense (Dense)               (None, 512)               205312

 dropout_1 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 9)                 4617

=================================================================
Total params: 256,354
Trainable params: 256,054
Non-trainable params: 300
_________________________________________________________________
```

```
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate=0.01), metrics=['
```

```
learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy', patience = 2, verbose=1,f
```

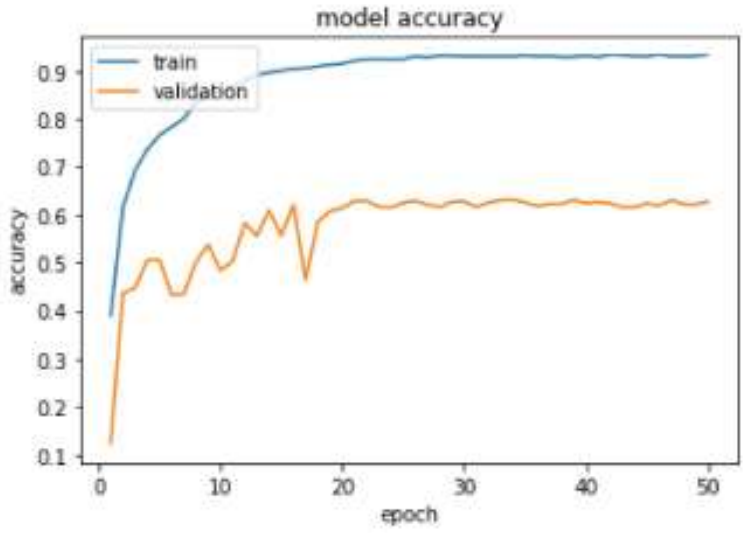
```
track=model.fit(
    x_train,
    steps_per_epoch=351,
    epochs=50,
    validation_data=x_test,
    validation_steps=88,
    initial_epoch=0,
    # callbacks=[LearningRateScheduler(adjustLearningRate)]
    callbacks=[learning_rate_reduction]
    )
```

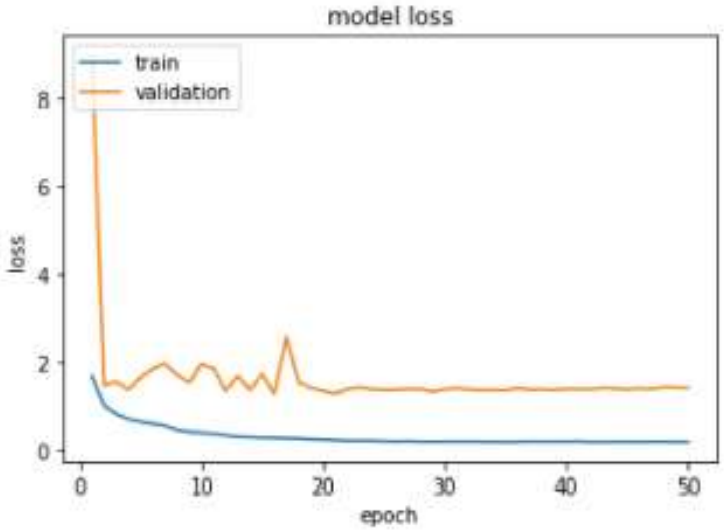
```
Epoch 27/50
351/351 [==============================] - 181s 517ms/step - loss: 0.2152 - accuracy:
Epoch 28/50
351/351 [==============================] - ETA: 0s - loss: 0.2064 - accuracy: 0.9319
Epoch 28: ReduceLROnPlateau reducing learning rate to 7.812499825377017e-05.
351/351 [==============================] - 181s 516ms/step - loss: 0.2064 - accuracy:
Epoch 29/50
351/351 [==============================] - 181s 514ms/step - loss: 0.2033 - accuracy:
Epoch 30/50
351/351 [==============================] - ETA: 0s - loss: 0.2078 - accuracy: 0.9304
Epoch 30: ReduceLROnPlateau reducing learning rate to 3.9062499126885086e-05.
351/351 [==============================] - 181s 516ms/step - loss: 0.2078 - accuracy:
Epoch 31/50
351/351 [==============================] - 181s 517ms/step - loss: 0.2073 - accuracy:
Epoch 32/50
351/351 [==============================] - ETA: 0s - loss: 0.2071 - accuracy: 0.9304
Epoch 32: ReduceLROnPlateau reducing learning rate to 1.9531249563442543e-05.
351/351 [==============================] - 181s 516ms/step - loss: 0.2071 - accuracy:
Epoch 33/50
351/351 [==============================] - 181s 514ms/step - loss: 0.2049 - accuracy:
Epoch 34/50
351/351 [==============================] - 181s 516ms/step - loss: 0.2016 - accuracy:
```

```
Epoch 35/50
351/351 [==============================] - 180s 511ms/step - loss: 0.1968 - accuracy:
Epoch 36/50
351/351 [==============================] - ETA: 0s - loss: 0.2028 - accuracy: 0.9303
Epoch 36: ReduceLROnPlateau reducing learning rate to 1e-05.
351/351 [==============================] - 185s 527ms/step - loss: 0.2028 - accuracy:
Epoch 37/50
351/351 [==============================] - 183s 520ms/step - loss: 0.2103 - accuracy:
Epoch 38/50
351/351 [==============================] - 182s 518ms/step - loss: 0.2057 - accuracy:
Epoch 39/50
351/351 [==============================] - 180s 514ms/step - loss: 0.2059 - accuracy:
Epoch 40/50
351/351 [==============================] - 180s 514ms/step - loss: 0.2061 - accuracy:
Epoch 41/50
351/351 [==============================] - 182s 518ms/step - loss: 0.2117 - accuracy:
Epoch 42/50
351/351 [==============================] - 179s 511ms/step - loss: 0.1979 - accuracy:
Epoch 43/50
351/351 [==============================] - 179s 508ms/step - loss: 0.2006 - accuracy:
Epoch 44/50
351/351 [==============================] - 180s 512ms/step - loss: 0.2009 - accuracy:
Epoch 45/50
351/351 [==============================] - 181s 516ms/step - loss: 0.2061 - accuracy:
Epoch 46/50
351/351 [==============================] - 181s 515ms/step - loss: 0.2005 - accuracy:
Epoch 47/50
351/351 [==============================] - 180s 512ms/step - loss: 0.2027 - accuracy:
Epoch 48/50
351/351 [==============================] - 180s 513ms/step - loss: 0.2036 - accuracy:
Epoch 49/50
351/351 [==============================] - 178s 507ms/step - loss: 0.2015 - accuracy:
Epoch 50/50
351/351 [==============================] - 179s 510ms/step - loss: 0.1979 - accuracy:
```

```
plt.figure(1)
plt.plot([1+i for i in track.epoch],track.history['accuracy'])
plt.plot([1+i for i in track.epoch],track.history['val_accuracy'])
# plt.xlim((1,5))
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()

plt.figure(2)
plt.plot([1+i for i in track.epoch],track.history['loss'])
plt.plot([1+i for i in track.epoch],track.history['val_loss'])
# plt.xlim((1,5))
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
```

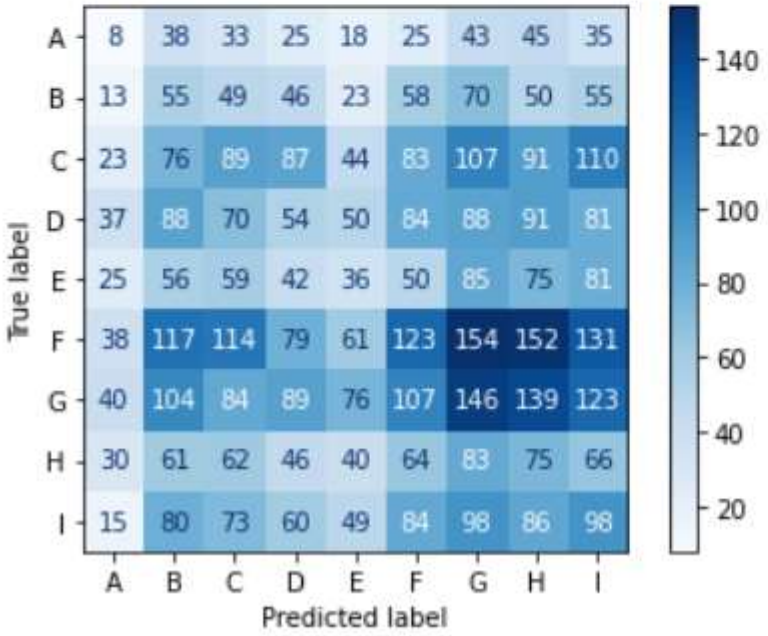
```
plt.legend(['train', 'validation'], loc='upper left')
plt.show()
```





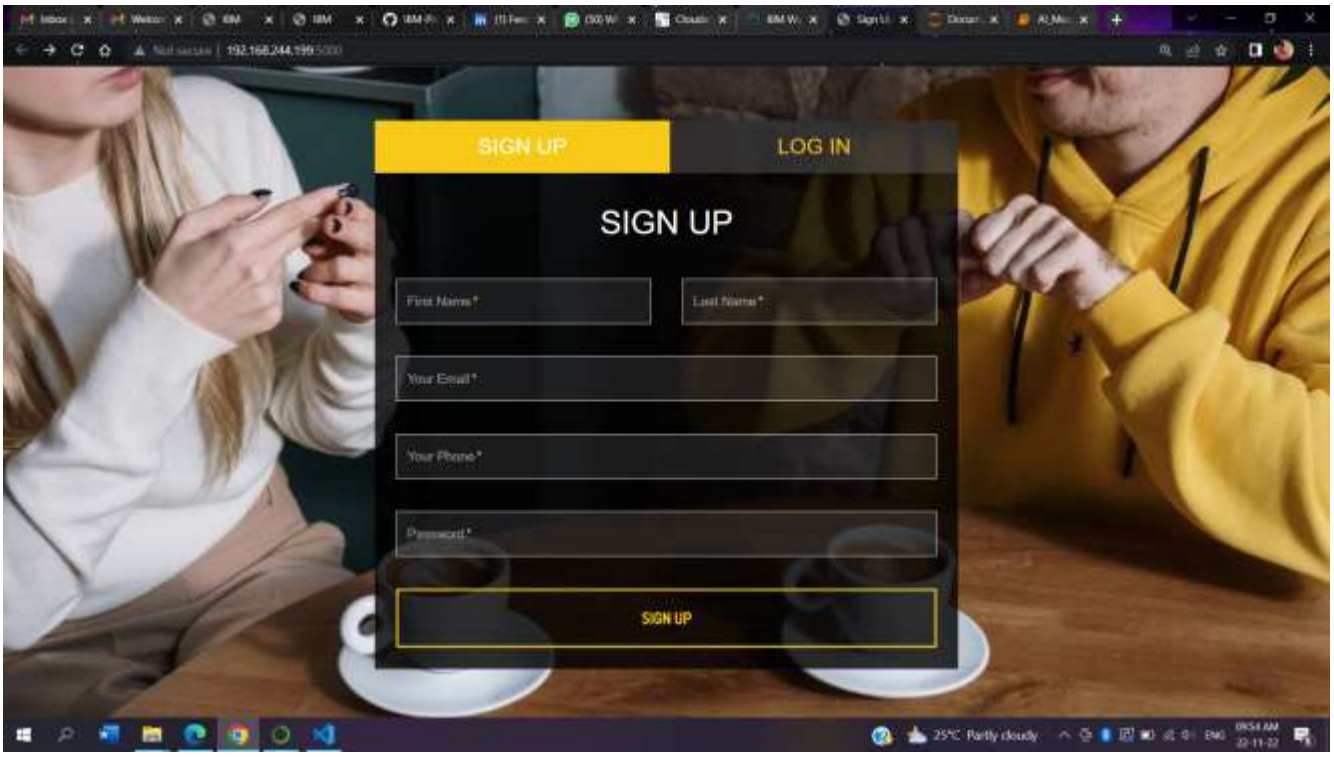

```
y_pred = model.predict(x_test, x_test.samples // x_test.batch_size+1)
y_pred = np.argmax(y_pred, axis=1)
confmat=confusion_matrix(x_test.classes, y_pred)
labels = ['A','B','C','D','E','F','G','H','I']
classrep=classification_report(x_test.classes, y_pred, target_names=labels)
print('Classification Report')
print(classrep)
confmat_show=ConfusionMatrixDisplay(confusion_matrix=confmat, display_labels=labels)
confmat_show.plot(cmap=plt.cm.Blues)
plt.show()
```

```
88/88 [==============================] - 28s 318ms/step
Classification Report
              precision    recall  f1-score   support

           A       0.03      0.03      0.03       270
           B       0.08      0.13      0.10       419
           C       0.14      0.13      0.13       710
           D       0.10      0.08      0.09       643
           E       0.09      0.07      0.08       509
           F       0.18      0.13      0.15       969
           G       0.17      0.16      0.16       908
           H       0.09      0.14      0.11       527
           I       0.13      0.15      0.14       643

    accuracy                           0.12      5598
   macro avg       0.11      0.11      0.11      5598
weighted avg       0.13      0.12      0.12      5598
```




```
model.save("weight17.h5")
```

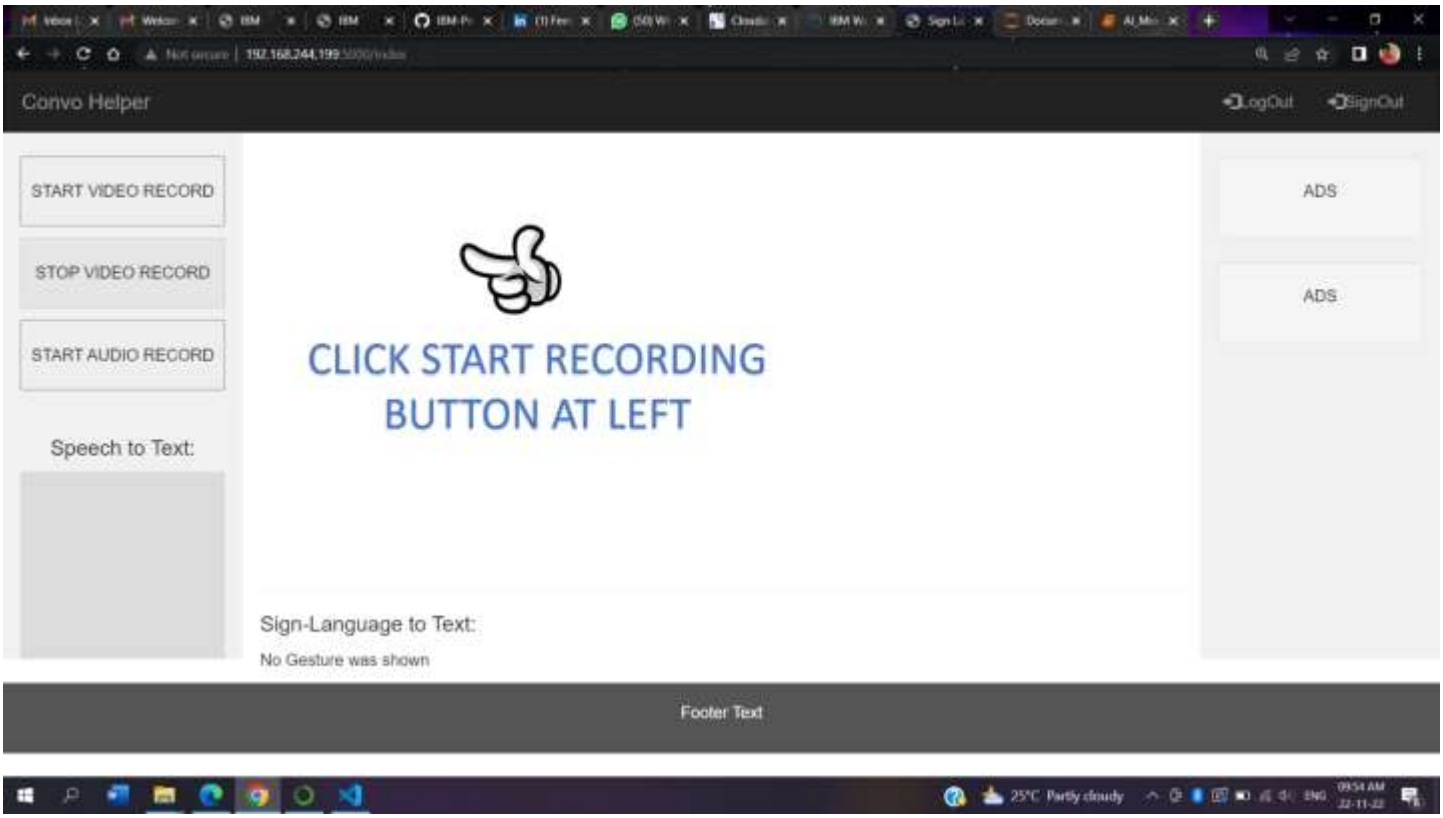
7.2 LOGIN AND REGISTER




```
@app.route('/signup', methods =["GET", "POST"])
def sign_up():
    global db_name, client, rev, doc_id
    if request.method == "POST":
        doc_id = request.form.get('mail')
        document: Document = Document(id=doc_id)
        document.fname = request.form.get("fname")
        document.lname = request.form.get("lname")
        document.phone=request.form.get('phone')
        password=request.form.get('password')
        document.password=hashlib.sha256(password.encode()).hexdigest()

        document_response =
client.post_document(db=db_name,document=document).get_result()
        document.rev = document_response["rev"]
        rev = document_response["rev"]
    return redirect(url_for('index'))

@app.route('/login', methods =["GET", "POST"])
def login():
```

```
    global db_name, client, error, doc_id
    if request.method == "POST":
        doc_id = request.form.get('mail')
        password=request.form.get('password')
        try:
            document = client.get_document(
                db=db_name,
                doc_id=doc_id
            ).get_result()
        except:
            error="Account not Registered. SignUp Here"
            return render_template('login_signup.html', error=error)
        cipher_password=hashlib.sha256(password.encode()).hexdigest()
        if document['password']==cipher_password:
            error=""
            return redirect(url_for('index'))
        else:
            # TODO
            error = "Password is Wrong"
            return render_template('login_signup.html', error=error)

@app.route('/signout', methods =["GET", "POST"])
def signout():
    global db_name, client, error, rev, doc_id
    document = client.delete_document(
        db=db_name,
        doc_id=doc_id,
        rev=rev
    ).get_result()
    return redirect(url_for('login_signup'))

@app.route('/logout', methods =["GET", "POST"])
def logout():
    return redirect(url_for('login_signup'))
```

7.3 VIDEO TO TEXT AND VOICE TO TEXT



Web App UPI

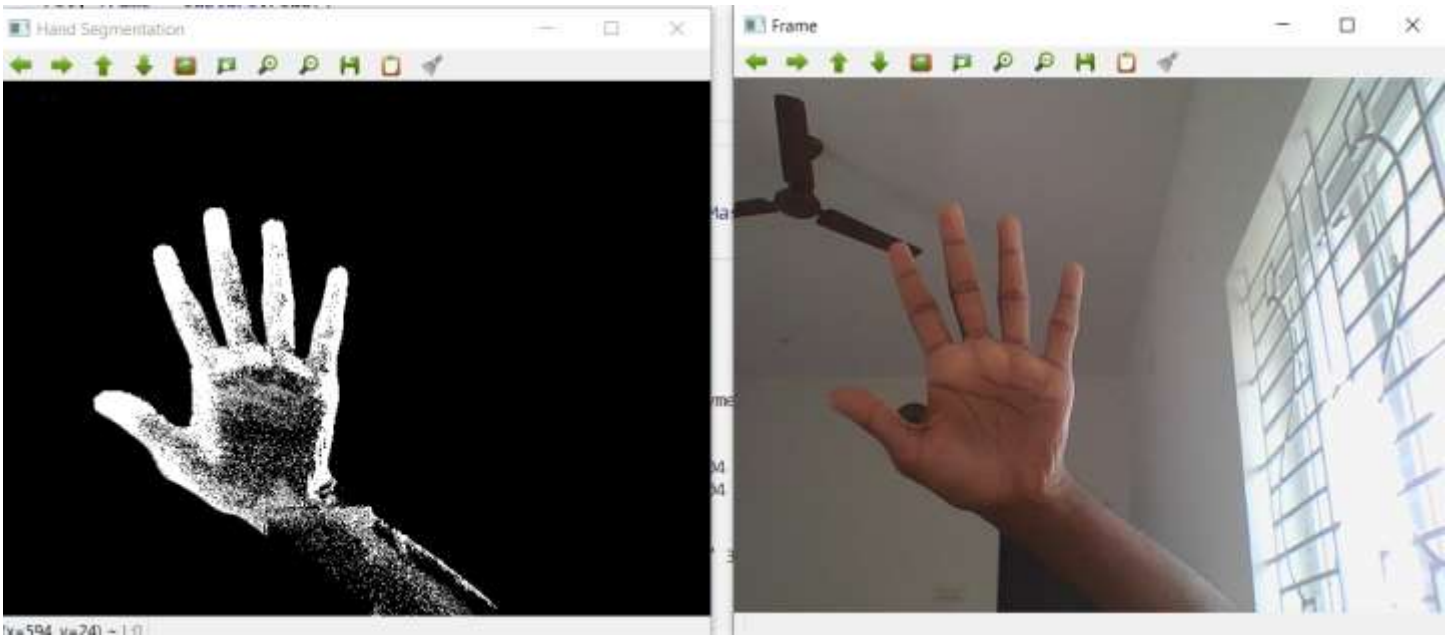


Image Segmentation

```
app=Flask(__name__)
# jsglue = JSGlue(app)
speechfile=0
audiooutput=0
lock_audio=False
transcript=""
pred_text = "No Gesture was shown"
error=""

def initialize_db():
    global db_name, client
    authenticator =
IAMAuthenticator('YIw8TkFJzxfUbZya9rp4YLHBVl3_HHrHk2cnDYjskbkG')
    client = CloudantV1(
        authenticator=authenticator
    )
    client.set_service_url('https://cf4faf1c-2967-4e3a-a69a-209a3c28cc3c-
bluemix.cloudantnosqldb.appdomain.cloud')
    db_name = "usersdb"

def initialize_speechToText():
    global speech_to_text
    authenticator = IAMAuthenticator('MGU3du2-
gL3o1rTmURlh2UHKkqM8AHnZqtHF1rg5P7pF')
    speech_to_text = SpeechToTextV1(
        authenticator=authenticator
    )
    speech_to_text.set_service_url('https://api.us-south.speech-to-
text.watson.cloud.ibm.com/instances/24675f56-6aa0-4aaf-abab-e5f8f38af20b')

@app.route('/speech', methods =["GET", "POST"])
def speechToText():
    global speech_to_text, lock_audio, transcript, speechfile
    lock_audio=True
    speechfile+=1
    freq = 44100
    duration = 5
    recording = sd.rec(int(duration * freq),samplerate=freq, channels=2)
    sd.wait()
    write("Application/audiofiles/audio"+str(speechfile)+".wav", freq, recording)
```

```
    with open("Application/audiofiles/audio"+str(speechfile)+".wav", 'rb') as audio_file:
        speech_recog=speech_to_text.recognize(
            audio=audio_file,
            content_type='audio/wav',
            model='en-US_Telephony'
        ).get_result()
    results=speech_recog['results']
    transcript=""
    for i in results:
        transcript+=" "+i['alternatives'][0]['transcript']
    print(transcript)
    lock_audio=False
    return redirect(url_for('index'))

def initialize_model():
    global graph,writer,model,vals,cap,pred,rec, backSub
    rec=False
    backSub = cv2.createBackgroundSubtractorKNN()
    graph=tf.compat.v1.get_default_graph()
    writer=None
    # with graph.as_default():
    model = load_model('Application/weight15.h5')
    vals=['A','B','C','D','E','F','G','H','I']
    print("[INFO] accessing video stream...")
    cap=cv2.VideoCapture(0)
    pred=" "
```

7.4 DATABASE SCHEMA

CHAPTER 8

TESTING

8.1 USER ACCEPTANCE TESTING

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	8	6	3	27
Duplicate	1	0	1	0	2
External	5	3	0	1	9
Fixed	9	1	2	15	27
Not Reproduced	0	0	0	1	1
Skipped	0	1	0	1	2
Won't Fix	0	4	0	1	5
Totals	25	17	9	22	73

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested.

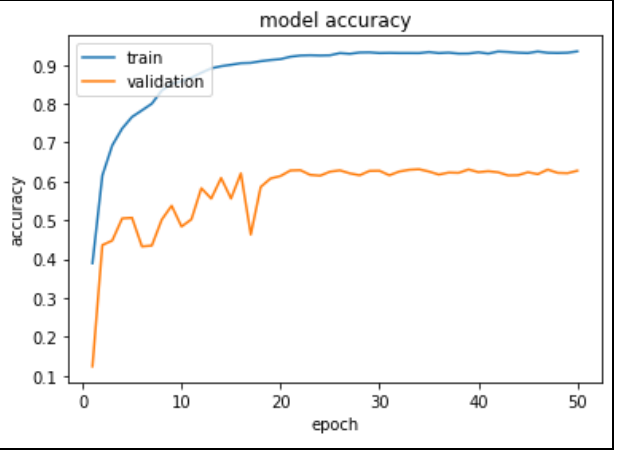
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application - Structural	10	0	0	10
Client Application - Functional	23	0	0	23
Security	3	0	0	3
Exception Reporting	5	0	0	5
Final Report Output	4	0	2	2
Version Control	1	0	0	1

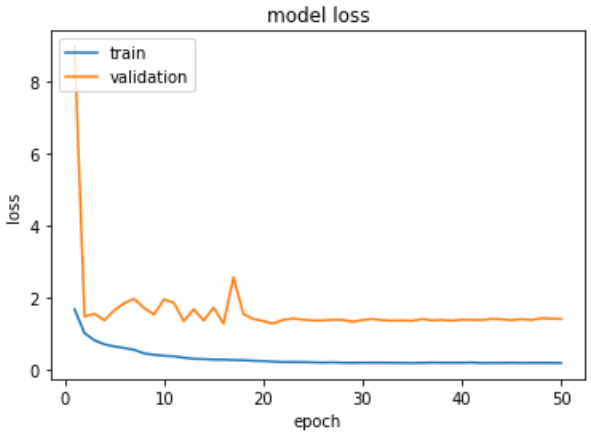
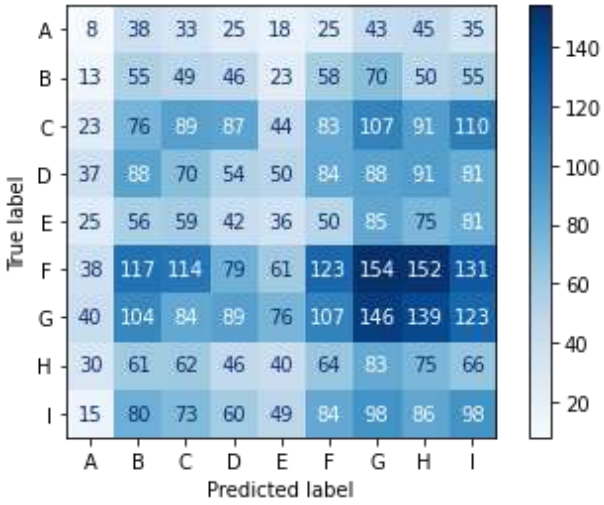
CHAPTER 9

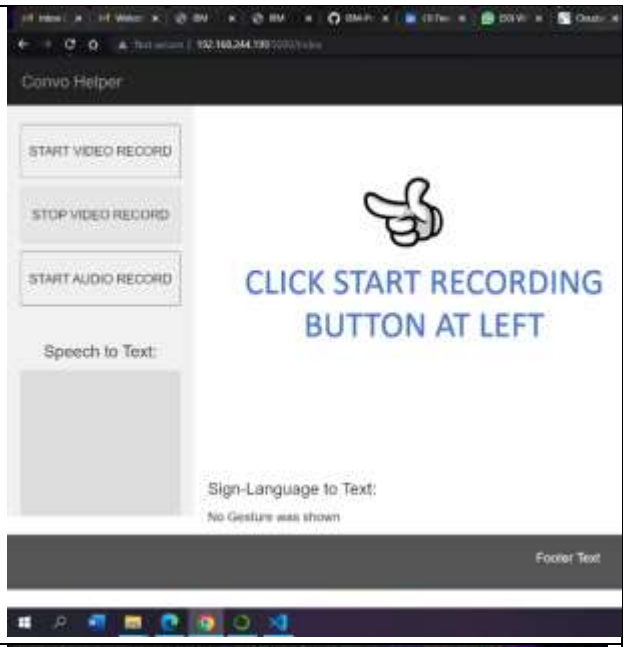
RESULTS

9.1 PERFORMANCE METRICS

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	Model: "sequential" _____ Layer (type) Output Shape Param # ===== conv2d (Conv2D) (None, 28, 28, 75) 750 batch_normalization (BatchN ormalization) (None, 28, 28, 75) 300 max_pooling2d (MaxPooling2D) (None, 14, 14, 75) 0 conv2d_1 (Conv2D) (None, 14, 14, 50) 33800 dropout (Dropout) (None, 14, 14, 50) 0 batch_normalization_1 (Batc hNormalization) (None, 14, 14, 50) 200 max_pooling2d_1 (MaxPooling 2D) (None, 7, 7, 50) 0 conv2d_2 (Conv2D) (None, 7, 7, 25) 11275 batch_normalization_2 (Batc hNormalization) (None, 7, 7, 25) 100 max_pooling2d_2 (MaxPooling 2D) (None, 4, 4, 25) 0 flatten (Flatten) (None, 400) 0 dense (Dense) (None, 512) 205312 dropout_1 (Dropout) (None, 512) 0 dense_1 (Dense) (None, 9) 4617 ===== Total params: 256,354 Trainable params: 256,054 Non-trainable params: 300 _____
2.	Accuracy	Training Accuracy – 0.9347 Validation Accuracy – 0.6275	

			
3.	Metrics	Confusion Matrix & Classification Report	 88/88 [==============================] - 28s 318ms/step Classification Report precision recall f1-score support A 0.03 0.03 0.03 270 B 0.08 0.13 0.10 419 C 0.14 0.13 0.13 710 D 0.10 0.08 0.09 643 E 0.09 0.07 0.08 509 F 0.18 0.13 0.15 969 G 0.17 0.16 0.16 908 H 0.09 0.14 0.11 527 I 0.13 0.15 0.14 643 accuracy 0.12 5598 macro avg 0.11 0.11 0.11 5598 weighted avg 0.13 0.12 0.12 5598

4	Dashboard Design		
5	Login/Register		

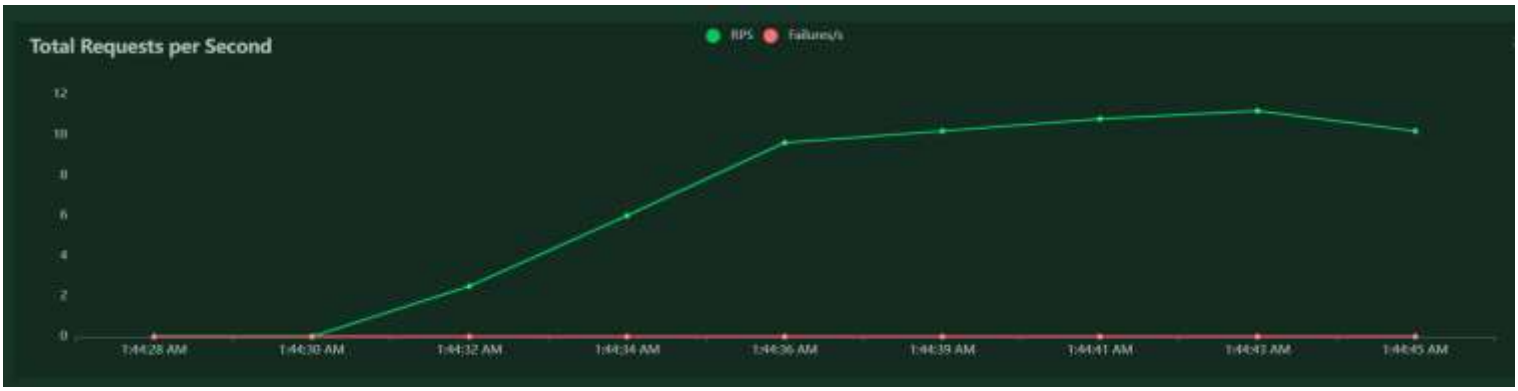
			NFT - Risk Assessment					
S.No	Project Name	Scope/ feature	Functional Changes	Hardware Changes	Software Changes	Load/Volume Changes	Risk Score	Justification
1	REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED	New	Moderate	No Changes	Moderate	>10 to 30%	GREEN	As we have seen the changes

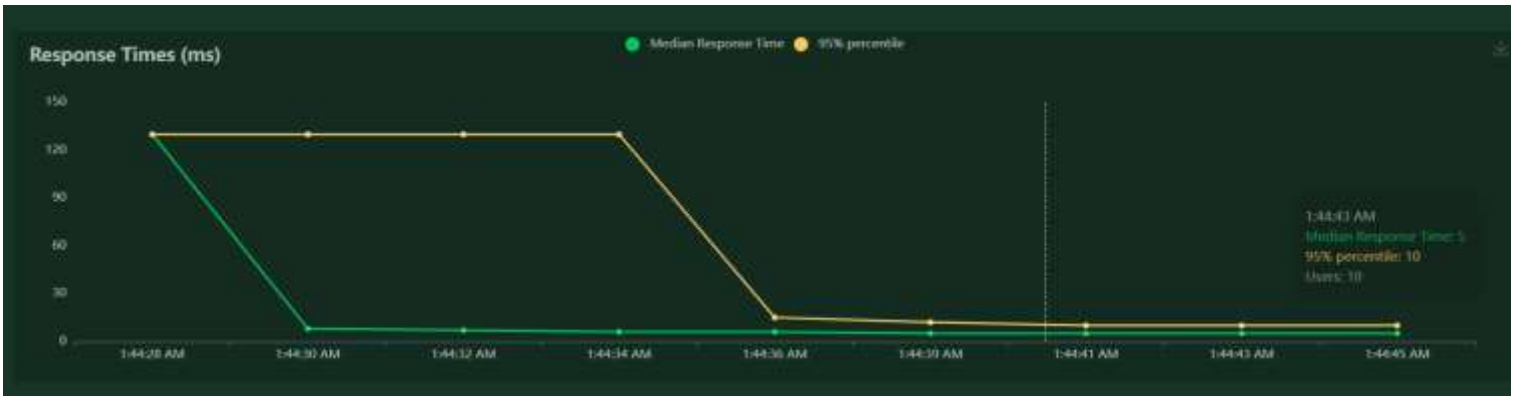
Performance testing using LOCUST

Statistics

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	/login	90	0	6	13	16	7	3	16	5317	5	0
GET	/signup	7	0	10	11	11	9	5	11	3647	0.4	0
GET	/video	99	0	4	9	130	10	2	133	0	4.8	0
	Aggregated	**196**	**0**	**5**	**11**	**130**	**9**	**2**	**133**	**2572**	**10.2**	**0**

Charts





Test Report

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/login	90	0	7	2	15	5317	4.9	0.0
GET	/signup	7	0	9	5	11	3647	0.4	0.0
GET	/video	99	0	10	2	133	0	5.4	0.0
	Aggregated	**196**	**0**	**8**	**2**	**133**	**2571**	**10.7**	**0.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/login	7	7	8	10	13	15	16	16
GET	/signup	10	10	10	10	11	11	11	11
GET	/video	4	5	6	7	9	12	130	130
	Aggregated	**6**	**6**	**7**	**9**	**11**	**14**	**130**	**130**

FINAL RATIO

RATIO PER USER CLASS

100.0% RealTime

- 2.4% signup_test
- 48.8% login_page_test
- 48.8% record_test

TOTAL RATIO

100.0% RealTime

- 2.4% signup_test
- 48.8% login_page_test
- 48.8% record_test

CHAPTER 10
ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- This project is a very cost-effective way of providing a simple solution to the targeted people
- It is a completely effective and reliable system
- The simplicity of usage is the main key here for users
- Utility is high and maintainability is low
- Host device is portable mobile which is available with everyone

DISADVANTAGES

- Storage needed will be more on storage of media
- Pre-requisite of using the app and conveying the knowledge to non-mute person can be difficult

CHAPTER 11

CONCLUSION

This project has been built on Convolutional Neural Networks with more tunings of parameters to achieve the maximum accuracy possible. Then using Flask, an application has been built as an interface for targeted people. UI will make usage easier, more protective, and more interactive. A login and sign-up page have been created for authorization and authentication purposes. “Anytime, Anywhere” users can sign in through any device. After login, users are provided with 2 features. First, video can be recorded and submitted after which the model will process the video and produce the output as text. Secondly, audio can be recorded and same as before, the text will be outputted for the given audio input. This system is therefore implemented in order for normal people to understand all sign languages and hand gestures used by deaf and mute people.

CHAPTER 12
FUTURE SCOPE

Initially, a simple UI and features were created for testing purposes. A few more features like converting text to speech using speech recognition, and converting voice to sign language for deaf people can be added to the project. The model's accuracy can be improved well to provide more accurate results with low loss and negligible errors. There are many types of sign languages used across different countries. More sign languages can be added to the app so that diversity can be increased. Customization of languages shall be added.

CHAPTER 13
APPENDIX

SOURCE CODE

Application Code –

```
import numpy as np
import cv2
import os
import hashlib
import tensorflow as tf
from keras.models import load_model
from flask import Flask, render_template, Response, request, redirect, url_for
# from flask_jsglue import JSGlue
from gtts import gTTS
from playsound import playsound
from skimage.transform import resize
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from ibmcloudant.cloudant_v1 import CloudantV1, Document
from ibm_watson import SpeechToTextV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
import sounddevice as sd
from scipy.io.wavfile import write
import wavio as wv
# from jinja2 import M

app=Flask(__name__)
# jsglue = JSGlue(app)
speechfile=0
audiooutput=0
lock_audio=False
transcript=""
pred_text = "No Gesture was shown"
error=""

def initialize_db():
    global db_name, client
    authenticator = IAMAuthenticator('YIw8TkFJzxfUbZya9rp4YLHBVl3_HHrHk2cnDYjskbkG')
    client = CloudantV1(
        authenticator=authenticator
    )
```

```
    client.set_service_url('https://cf4faf1c-2967-4e3a-a69a-209a3c28cc3c-
bluemix.cloudantnosqldb.appdomain.cloud')
    db_name = "usersdb"

def initialize_speechToText():
    global speech_to_text
    authenticator = IAMAuthenticator('MGU3du2-
gL3o1rTmURlh2UHKkqM8AHnZqtHF1rg5P7pF')
    speech_to_text = SpeechToTextV1(
        authenticator=authenticator
    )
    speech_to_text.set_service_url('https://api.us-south.speech-to-
text.watson.cloud.ibm.com/instances/24675f56-6aa0-4aaf-abab-e5f8f38af20b')

# def textToSpeech():
#     global speechfile
#     speechfile+=1
#     freq = 44100
#     duration = 10
#     recording = sd.rec(int(duration * freq),samplerate=freq, channels=2)
#     sd.wait()
#     write("audiofiles/audio"+str(speechfile)+".wav", freq, recording)

@app.route('/speech', methods =["GET", "POST"])
def speechToText():
    global speech_to_text, lock_audio, transcript, speechfile
    lock_audio=True
    speechfile+=1
    freq = 44100
    duration = 5
    recording = sd.rec(int(duration * freq),samplerate=freq, channels=2)
    sd.wait()
    write("Application/audiofiles/audio"+str(speechfile)+".wav", freq, recording)
    with open("Application/audiofiles/audio"+str(speechfile)+".wav", 'rb') as audio_file:
        speech_recog=speech_to_text.recognize(
            audio=audio_file,
            content_type='audio/wav',
            model='en-US_Telephony'
        ).get_result()
    results=speech_recog['results']
```

```
    transcript=""
    for i in results:
        transcript+=" "+i['alternatives'][0]['transcript']
    print(transcript)
    lock_audio=False
    return redirect(url_for('index'))

def initialize_model():
    global graph,writer,model,vals,cap,pred,rec, backSub
    rec=False
    backSub = cv2.createBackgroundSubtractorKNN()
    graph=tf.compat.v1.get_default_graph()
    writer=None
    # with graph.as_default():
    model = load_model('Application/weight15.h5')
    vals=['A','B','C','D','E','F','G','H','I']
    print("[INFO] accessing video stream...")
    cap=cv2.VideoCapture(0)
    pred=""

@app.route('/signup', methods =["GET", "POST"])
def sign_up():
    global db_name, client, rev, doc_id
    if request.method == "POST":
        doc_id = request.form.get('mail')
        document: Document = Document(id=doc_id)
        document.fname = request.form.get("fname")
        document.lname = request.form.get("lname")
        document.phone=request.form.get('phone')
        password=request.form.get('password')
        document.password=hashlib.sha256(password.encode()).hexdigest()

        document_response =
client.post_document(db=db_name,document=document).get_result()
        document.rev = document_response["rev"]
        rev = document_response["rev"]
    return redirect(url_for('index'))

@app.route('/login', methods =["GET", "POST"])
def login():
```

```
    global db_name, client, error, doc_id
    if request.method == "POST":
        doc_id = request.form.get('mail')
        password=request.form.get('password')
        try:
            document = client.get_document(
                db=db_name,
                doc_id=doc_id
            ).get_result()
        except:
            error="Account not Registered. SignUp Here"
            return render_template('login_signup.html', error=error)
        cipher_password=hashlib.sha256(password.encode()).hexdigest()
        if document['password']==cipher_password:
            error=""
            return redirect(url_for('index'))
        else:
            # TODO
            error = "Password is Wrong"
            return render_template('login_signup.html', error=error)

@app.route('/signout', methods =["GET", "POST"])
def signout():
    global db_name, client, error, rev, doc_id
    document = client.delete_document(
        db=db_name,
        doc_id=doc_id,
        rev=rev
    ).get_result()
    return redirect(url_for('login_signup'))

@app.route('/logout', methods =["GET", "POST"])
def logout():
    return redirect(url_for('login_signup'))

def detect(frame):
    global pred_text, audiooutput
    # print("Inside Detect")
    # cv2.imshow('frame', frame)
    mask = backSub.apply(frame)
```

```
    img = resize(mask, (28,28,1))
    img = np.expand_dims(img, axis=0)
    if np.max(img) > 1:
        img = img/255
    prediction = model.predict(img)
    if len(np.where(prediction[0]>=0.5)[0]) > 0:
        pred=vals[np.where(prediction[0]>=0.5)[0][0]]
        if pred_text == "No Gesture was shown":
            pred_text = ""
            pred_text = pred
        elif pred != pred_text[-1]:
            pred_text = pred_text + " " + pred

def gen():
    while cap.isOpened():
        ret, frame = cap.read()
        if ret:
            ret1, buffer = cv2.imencode('.png', frame)
            frame1 = buffer.tobytes()
            yield (b'--frame\r\n'
                   b'Content-Type: image/png\r\n\r\n' + frame1 + b'\r\n')
            detect(frame)
        else:
            exit
    cap.release()
    cv2.destroyAllWindows()

@app.route('/')
def login_signup():
    # render_template('static/script.js')
    return render_template('login_signup.html', error=error)
    # return render_template('index.html', var=rec,
transcript=transcript,lock_audio=lock_audio,pred_text=pred_text)

@app.route('/index')
def index():
    # render_template('static/script.js')
    return render_template('index.html',
var=rec,transcript=transcript,lock_audio=lock_audio,pred_text=pred_text)
```

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

@app.route('/record', methods =["GET", "POST"])
def record():
    global rec, cap, audiooutput
    if request.method == "POST":
        start_rec = request.form.get('start_rec')
        print(start_rec)
    if rec==False:
        rec=True
    else:
        rec=False
        cap.release()
        myobj = gTTS(text=pred_text, lang='en', slow=False)
        audiooutput+=1
        myobj.save("Application/speechoutput/speech"+str(audiooutput)+".mp3")
        playsound("Application/speechoutput/speech"+str(audiooutput)+".mp3")
    # print(rec)
    return redirect(url_for('index', var=rec))

if __name__=="__main__":
    initialize_db()
    initialize_speechToText()
    initialize_model()
    app.run(host='0.0.0.0')
```

GitHub Link: **https://github.com/IBM-EPBL/IBM-Project-13189-1659513642**

Project Demo Link:
https://drive.google.com/file/d/1nFrZ2TLuHVjP99s3F2WxKnFS2ciTawuW/view?usp=sharing