REAL – TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

A PROJECT REPORT

Submitted by

NAGENDRAPRASANTH A

KRISHNA RAJ R

SAI VIGNESH S

SIVA BALAJI P

ANANDARAJ K

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PARK COLLEGE OF ENGINEERING AND TECHNOLOGY

COIMBATORE 641659

ANNA UNIVERSITY :: CHENNAI 600 025

JANUARY 2023

PARK COLLEGE OF ENGINEERING AND TECHNOLOG

COIMBATORE 641659

ANNA UNIVERSITY :: CHENNAI 600 025

PROJECT VIVA-VOCE EXAMINATION

The Viva-Voce Examination of the project work submitted by,

NAGENDRAPRASANTH A	**712219104017**
KRISHNA RAJ R	**712219104013**
SAI VIGNESH S	**712219104020**
SIVA BALAJI P	**712219104024**
ANANDARAJ K	**712219104004**

is to be held on ..... at Computer Science and Engineering Department.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO.
1	**INTRODUCTION**	
	1.1 PROJECT OVERVIEW	1
	1.2 PURPOSE	2
2	**LITERATURE SURVEY**	
	2.1 EXISTING PROBLEM	3
	2.2 REFERENCES	4
	2.3 PROBLEM STATEMENT DEFINITION	5
3	**IDEATION & PROPOSED SOLUTION**	
	3.1 EMPATHY MAP CANVAS	6
	3.2 IDEATION &BRAINSTORMING	7
	3.3 PROPOSED SOLUTION	9
	3.4 PROBLEM SOLUTION FIT	10
4	**REQUIREMENT ANALYSIS**	
	4.1 FUNCTIONAL REQUIREMENTS	12
	4.2 NON-FUNCTIONAL REQUIREMENTS	12
5	**PROJECT DESIGN**	
	5.1 DATA FLOW DIAGRAMS	14
	5.2 SOLUTION & TECHNICAL ARCHITECTURE	15
	5.3 USER STORIES	17
6	**PROJECT PLANNING AND SCHEDULING**	
	6.1 SPRINT PLANNING &ESTIMATION	18
	6.2 SPRINT DELIVERY SCHEDULE	20
	6.3 REPORTS FROM JIRA	20
7	**CODING & SOLUTIONING**	
	7.1 FEATURE 1	21
	7.2 FEATURE 2	33

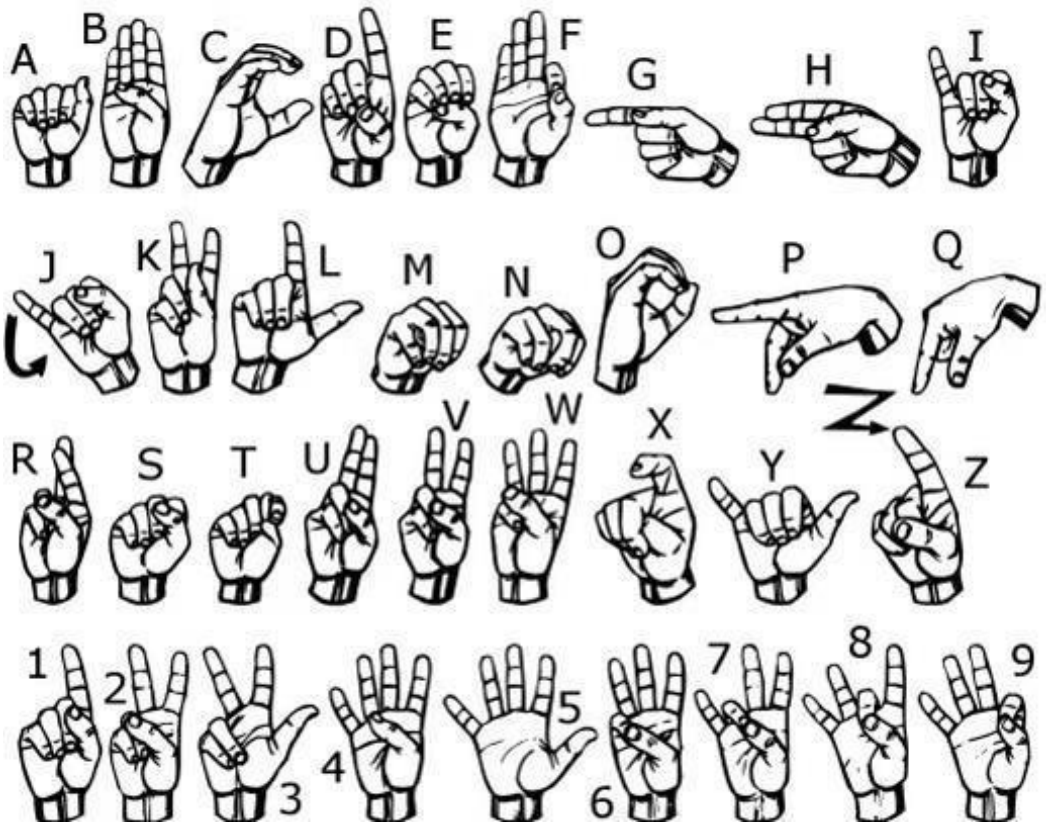
CHAPTER	TITLE	PAGE NO.
8	**TESTING**	
	8.1 TEST CASES	**35**
	8.2 USER ACCEPTANCE TESTING	**36**
9	**RESULTS**	
	9.1 PERFORMANCE METRICS	**37**
10	**ADVANTAGES & DISADVANTAGES**	**38**
11	**CONCLUSION**	**39**
12	**FUTURE SCOPE**	**40**
13	**APPENDIX**	**41**
	• SOURCE CODE	
	• GITHUB & PROJECT DEMO LINK	

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

In our daily life, the communication between several different communities is fundamental and very much important to share information. Being able to communicate effectively is a vital life skill but for the people with speech and hearing disability, they find it difficult to convey their messages with others. The process of communication between two people can be done using various medium. Not everyone knows how to interpret a sign language when having a conversation with such community like of deaf and dumb person. One finds it difficult to communicate without an interpreter or some other sources. We need to convert the sign language so that it is understood by others and also help them to communicate without any barriers. One of the effective solutions of this difficulty is sign language recognition system. Using sign language different gestures of hand are used to express meaningful information. Language of sign is different in different parts of world. There are 135 sign languages prevalent throughout the world for communication. Each sign language is different from the other like American Sign Language used in America is different from the Indian Sign Language of India. Looking to the ease of understanding Indian Sign Language, we standardized to work on Indian Sign Language gestures. We need to convert the sign language so that it is understood by others and also help them to communicate without any barriers. Sign language recognition is still a challenging problem inspire of many research efforts during the last many years. One of the methods of hand gesture recognition is to use the hand gloves for human computer interaction. But this method is sophisticated as it requires user to wear glove and carry a load of cables connecting the device to a computer.

Therefore, to eliminate this complication and to make user interaction with computer easy and natural we proposed to work on sign recognition using bare hands i.e., no usage of any external wearable hardware. Mainly sign language recognition processes are highly depending on human based translation services. The involvement of human expertise is very difficult and expensive also for translation. Now our proposed automatic sign language recognition system leads to understand the meaning of different signs without any aid from the expert. In common, any sign language recognition system contains several modules like object tracking, skin segmentation, feature extraction, and recognition. The first two modules



1.2 PURPOSE

The motivation behind building such a system includes:

- Sign-to-text translation system or dialog systems which can be used in various public domains like in Government Departments, airports etc.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

As per the journal on Hand Gesture Recognition Using data acquisition and preprocessing" by Dr. Dasaraju Srinivasa Rao constructed a skin model to extract the hand out of an image and then apply binary threshold to the whole image. After obtaining the threshold image they calibrate it about the principal axis in order to centre the image about it. He input this image to a convolutional neural network model in order to train and predict the outputs. He trained the model over 7 hand gestures and using the model he produced an accuracy of around 95% for those 7 gestures. Problem resulted in this survey is that it does not consider gesture recognition in temporal space and unable to classify the images in the complex background.

Hand Gesture Recognition using Kinetic Camera:

As per the title, produced by Dr. Partha Talukdar published on Journal of Inst. Of Engineers India : Series – A October based on classifying the different gestures according to geometric based invariants which are obtained from image data after segmentation. Thus, unlike many other recognition methods, this method is not dependent on skin colour. The gestures are extracted from each frame of the video, with a static background. The first step is to segment and label the objects of interest and to extract geometric invariants from them. Change in the illumination , rotation and orientation , scaling problem is a the drawback encountered.

On the Other hand, G.Pushpak Bhattacharya published a model dealing with the applications of AI in Machine Learning encountered drawback mainly on edge detection and segmentation algorithms used here are not very efficient.

2.2 REFERENCES

[1] Aman Aryan and Subham Sanghai, “Indian Sign Language Recognition by Feature Extraction Using SURF”, Birla Institute of Technology, Mesra.

[2] Beena M.V. and Dr. M.N. Agnisarman Namboodiri, “Automatic Sign Language Finger Spelling Using Convolution Neural Network: Analysis”, 2017 International Journal of Pure and Applied Mathematics.

[3] Hsien-I Liny, Ming-Hsiang Hsu, and Wei-Kai Chen, “Human Hand Gesture Recognition Using a Convolution Neural Network”, 2014 IEEE International Conference on Automation Science and Engineering (CASE).

[4] Brandon Garcia, Sigberto Alarcon Viesca, “Real-time American Sign Language Recognition with Convolutional Neural Networks”,2016 Stanford University Stanford, CA.

[5] Sirshendu Hore, Sankhadeep Chatterjee, V. Santhi and Nilanjan Dey, “Indian Sign Language Recognition using Optimized Neural Networks”, 2015 International Conference on Information Technology and Intelligent Transportation Systems.

[6] Mahesh Kumar N B, “Conversion of Sign Language into Text”, International Journal of Applied Engineering Research ISSN 0973 -4562 Volume 13, Number 9 (2018) pp. 7154-7161.

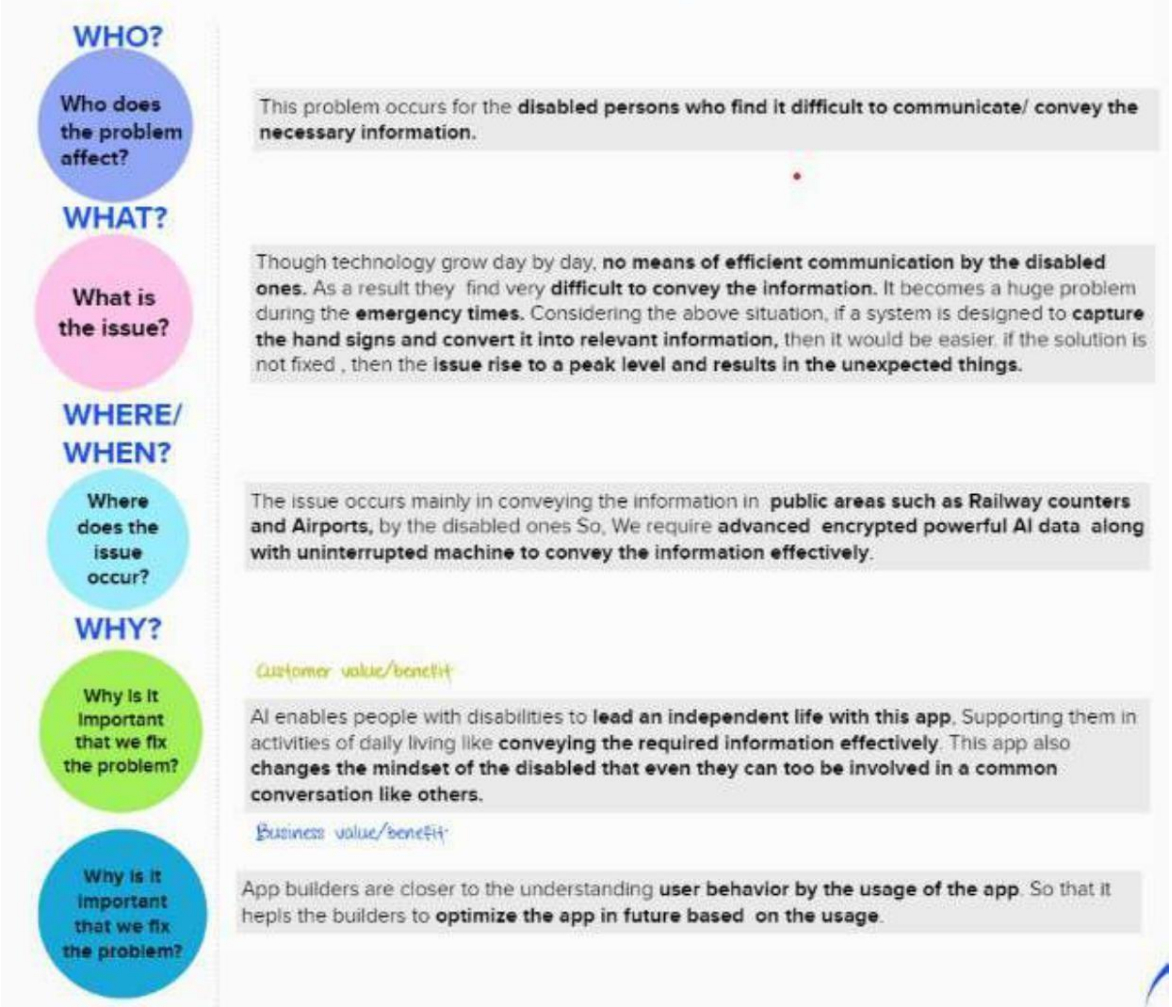
[7] Kuntal Kumar Pal, Sudeep K. S., “Pre-processing for Image Classification by Convolutional Neural Networks”, IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016, India.

2.3 PROBLEM STATEMENT DEFINITION

A problem statement is a concise description of an issue to be addressed or a condition to be improved upon. It identifies the gap between the current state and desired state of a process or product.

For the given project, the following set of questions are raised .

- WHO?
- WHAT?
- WHERE/WHEN
- WHY?

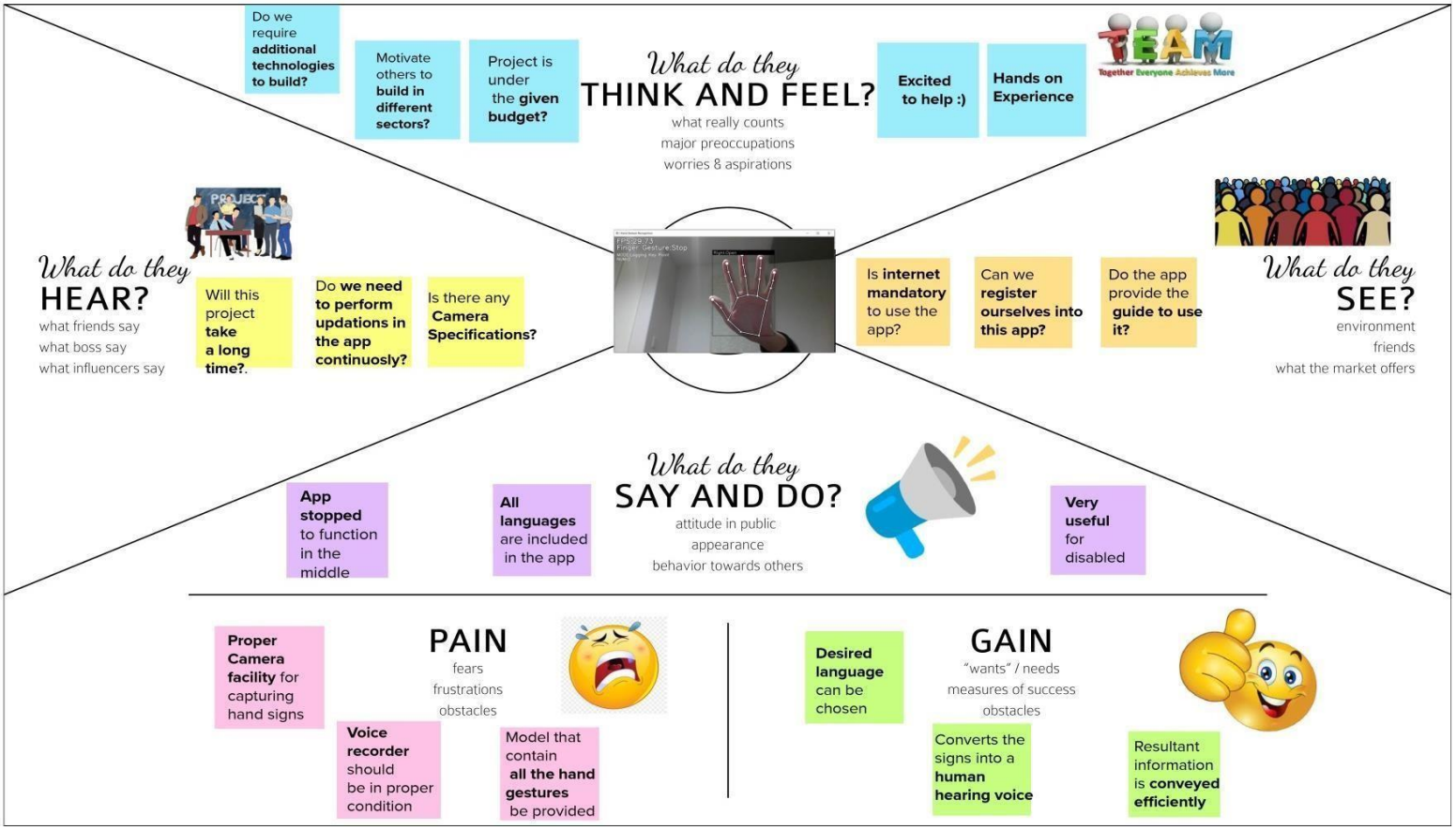


CHAPTER 3
IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An **empathy map** is a collaborative visualization used to articulate what we know about a particular type of user. Empathy maps should be used throughout any UX process to establish common ground among team members and to understand and prioritize user needs. In user-centered design, empathy maps are best used from the very **beginning of the design process**.It externalizes knowledge about users in order to

- Create a shared understanding of user needs,
- Aid in decision making.



3.2 IDEATION & BRAINSTORMING

Brainstorming combines a relaxed, informal approach to problem solving with lateral thinking. It encourages people to come up with thoughts and ideas that can, at first, seem a bit crazy. Some of these ideas can be crafted into original, creative solutions to a problem, while others can spark even more ideas. This helps to get people unstuck by "jolting" them out of their normal ways of thinking.

For the given Project, different ideas and methods were suggested in order to come up with a solution for the existing problem.

3.3 PROPOSED SOLUTION

The main objective of the project is to contribute to the field of automatic sign language recognition. Our focus is mainly on the recognition of the real time sign language gestures. This work focused on deep learning approach to numbers, alphabets and most often used words. Through this work we want to ease the interaction for people with speech and hearing disabilities and also other objective is to convert sign language into text.

Problem statement:

- To provide an Efficient communication app which translates the hand signs into text and voice mode for deaf and dumb people.

Idea / Solution description:

- Convolution Neural Networks are to be used to take hand sign as

an input to extract edges, corners.

- Dataset is used for training CNN. One dataset for hand detection and the other for gesture detection.
- Voice assistant is implemented that take input as speech patterns and convert the text into voice

Novelty / Uniqueness:

We have number of symbols to be trained for our project and many of them look similar to each other like the gesture for symbol ‘V’ and digit ‘2’ .To produce better accuracies, we keep the background of hand a stable single colour, so that we don’t need to segment it on basis of skin colour.

Social Impact / Customer Satisfaction:

- AI enables people with disabilities to lead an independent life with this app.
- Supporting them in activities of daily living .
- It changes the mind set of the disabled, that even they can to be involved in a common conversation like others.

Business Model (Revenue Model):

- Faster and efficient , the concerned text or voice as output is produced, the more it leads to optimize the app with new advancements.
- The productivity is gained and at the same time, leads to improved speed of business.

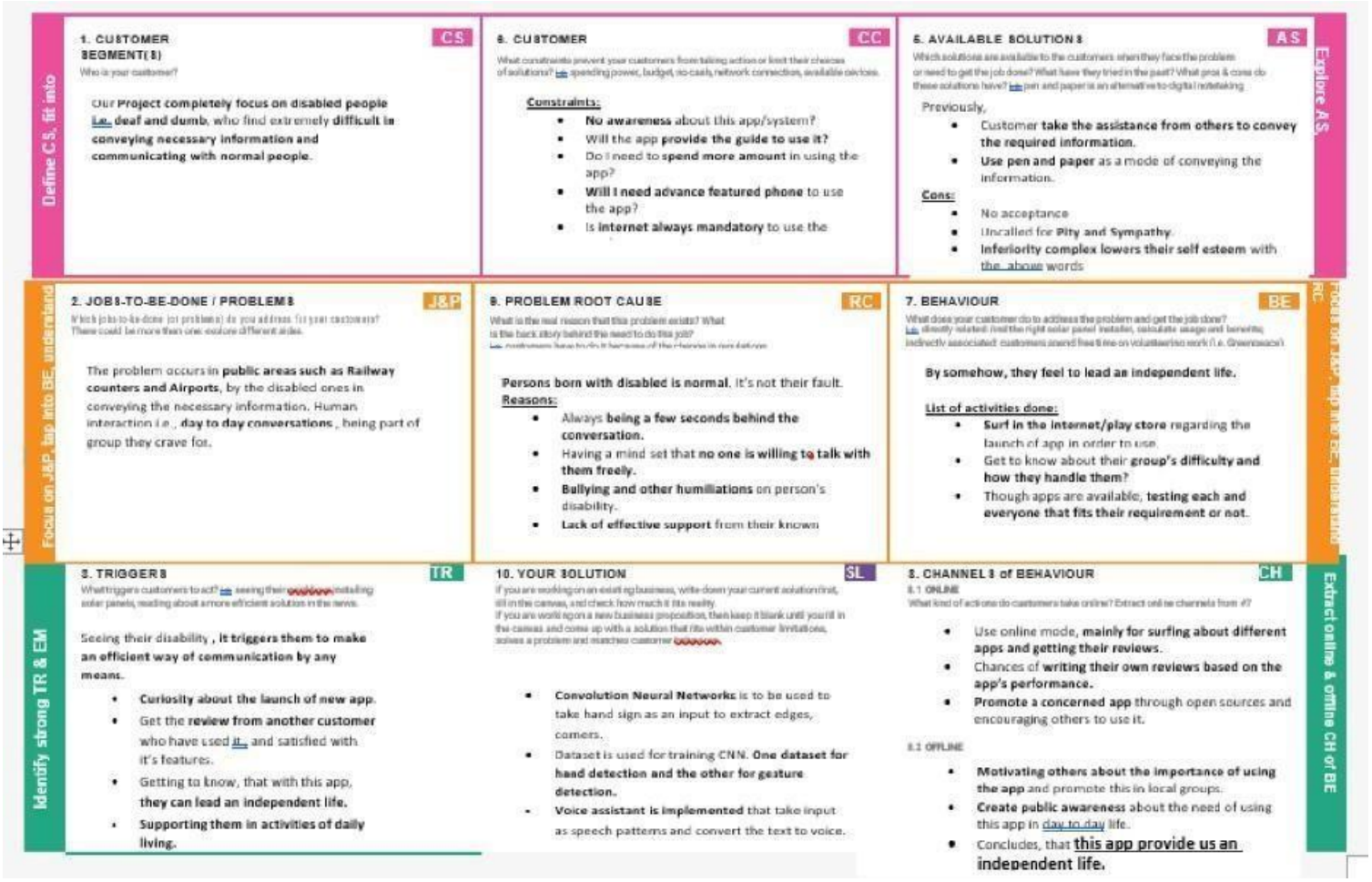
Scalability of the Solution:

- A convolutional neural network can be scaled in three dimensions: depth, width, resolution.
- Depth of the network corresponds to the number of layers in a network.
- Width is associated with the number of neurons in a layer.
- Resolution is the image resolution that is being passed to CNN. Increasing the depth, by stacking more convolutional layers, allows the network to learn more complex features.

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that we have found a problem with the customer and the solution we realized for it actually solves the customer's problem.

For the given project, the following situations are taken into consideration and at last proposed the innovative ideas in order to provide solution to the problem.


Define CS, fit into
1. CUSTOMER SEGMENT(S)
CS
Who is your customer?
Our Project completely focus on disabled people i.e., deaf and dumb, who find extremely difficult in conveying necessary information and communicating with normal people.
6. CUSTOMER CONSTRAINTS
CC
What constraints prevent your customers from taking action or limit their choices of solutions? i.e., spending power, budget, no cash, network connection, available devices.
Constraints:
No awareness about this app/system?
Will the app provide the guide to use it?
Do I need to spend more amount in using the app?
Will I need advance featured phone to use the app?
Is internet always mandatory to use the
5. AVAILABLE SOLUTIONS
AS
Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking
Previously,
Customer take the assistance from others to convey the required information.
Use pen and paper as a mode of conveying the information.
Cons:
No acceptance
Uncalled for Pity and Sympathy.
Inferiority complex lowers their self esteem with the above words
Explore AS,
Focus on J&P, tap into BE, understand
2. JOBS-TO-BE-DONE / PROBLEMS
J&P
Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.
The problem occurs in public areas such as Railway counters and Airports, by the disabled ones in conveying the necessary information. Human interaction i.e., day to day conversations, being part of group they crave for.
9. PROBLEM ROOT CAUSE
RC
What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.
Persons born with disabled is normal. It's not their fault.
Reasons:
Always being a few seconds behind the conversation.
Having a mind set that no one is willing to talk with them freely.
Bullying and other humiliations on person's disability.
Lack of effective support from their known
7. BEHAVIOUR
BE
What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)
By somehow, they feel to lead an independent life.
List of activities done:
Surf in the internet/play store regarding the launch of app in order to use.
Get to know about their group's difficulty and how they handle them?
Though apps are available, testing each and everyone that fits their requirement or not.
Find TR & EM, then focus on
RC
Identify strong TR & EM
3. TRIGGERS
TR
What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.
Seeing their disability, it triggers them to make an efficient way of communication by any means.
Curiosity about the launch of new app.
Get the review from another customer who have used it, and satisfied with it's features.
Getting to know, that with this app, they can lead an independent life.
Supporting them in activities of daily living.
10. YOUR SOLUTION
SL
If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer
Convolution Neural Networks is to be used to take hand sign as an input to extract edges, corners.
Dataset is used for training CNN. One dataset for hand detection and the other for gesture detection.
Voice assistant is implemented that take input as speech patterns and convert the text to voice.
8. CHANNELS of BEHAVIOUR
CH
8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7
Use online mode, mainly for surfing about different apps and getting their reviews.
Chances of writing their own reviews based on the app's performance.
Promote a concerned app through open sources and encouraging others to use it.
8.2 OFFLINE
Motivating others about the importance of using the app and promote this in local groups.
Create public awareness about the need of using this app in day to day life.
Concludes, that this app provide us an independent life.
Extract online & offline CH of BE

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

- Here , Desktop along with Camera is presented as black box.
- Deaf/Dumb is the person, who will show different signs based on the type of information being conveyed.
- Normal Person is the passive user of the desktop.

The System requirements that are required are specified below,

- Deaf/Dumb person should be able to perform a sign that represents digit/number.
- Deaf/Dumb person should be able to perform a sign that represents a character.
- Deaf/Dumb person should be able to perform a sign , where group of characters Forms a word.
- Deaf/Dumb person should be able to perform a sign, where group of words forms a sentence.
- Especially Deaf person should be able to see the translation of sign to text format.
- Dumb person should be able to understand the conversion of text into voice mode.
- Normal user should be able to understand the corresponding information conveyed by disabled through sign language.

4.2 NON-FUNCTIONAL REQUIREMENTS

Some of the non -functional requirements are mentioned as shown below.

- **SPEED:**

Speed determines how fast an application responds to commands. For example, it can be considered the amount of time , the concerned application gets opened once clicked.

- **COMPATIBILITY:**

  Highly compatible systems typically function well when other applications are running on a device. Compatibility also allows people who have different operating systems to use the same applications.

- **CAPACITY:**

  The capacity of a system refers to the amount of storage it offers. When using some applications, users can adjust and save settings based on their preferences.

- **RELIABILITY:**

  Technology that is highly reliable functions with the same or similar efficiency after extensive use

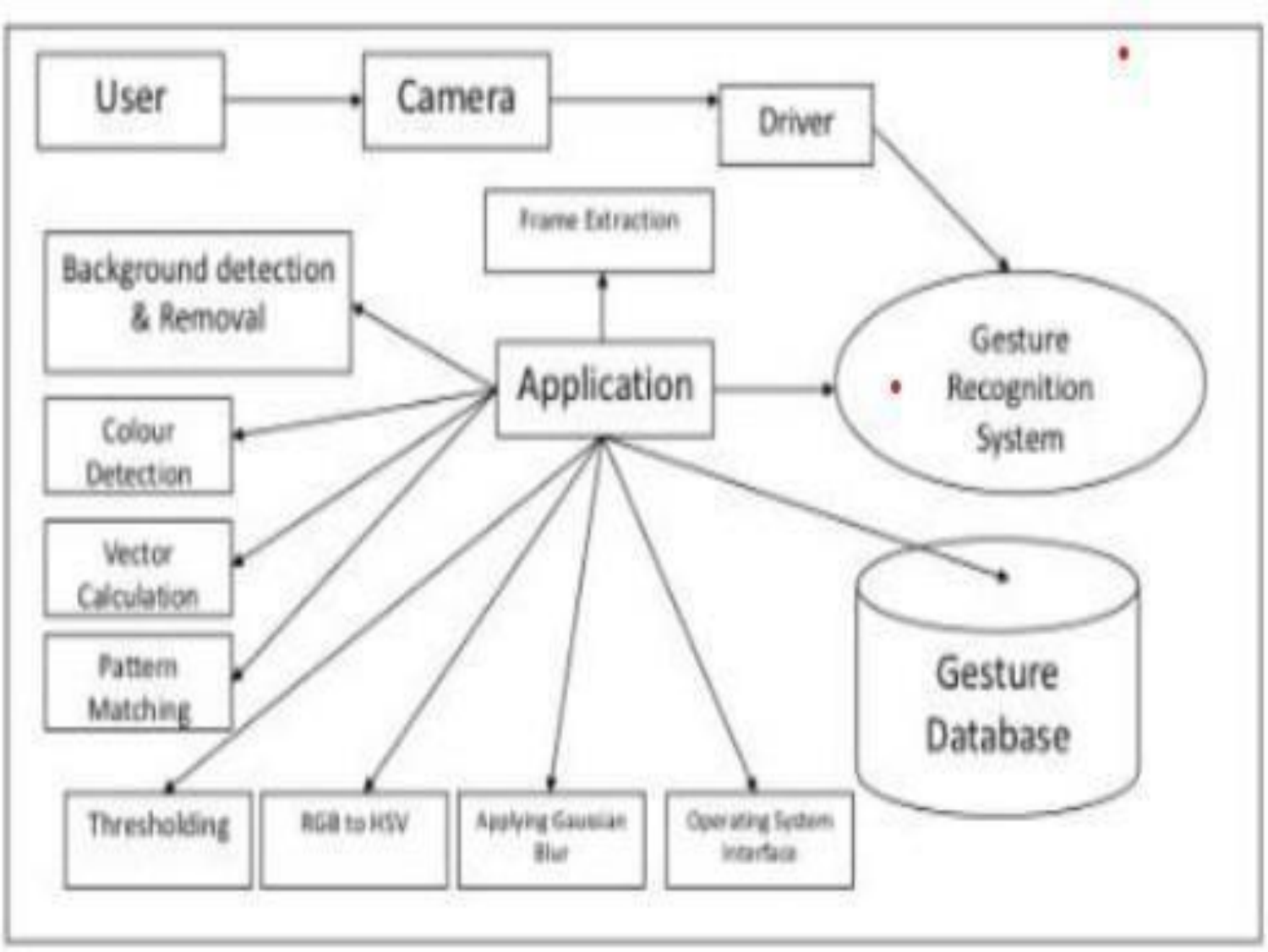
CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

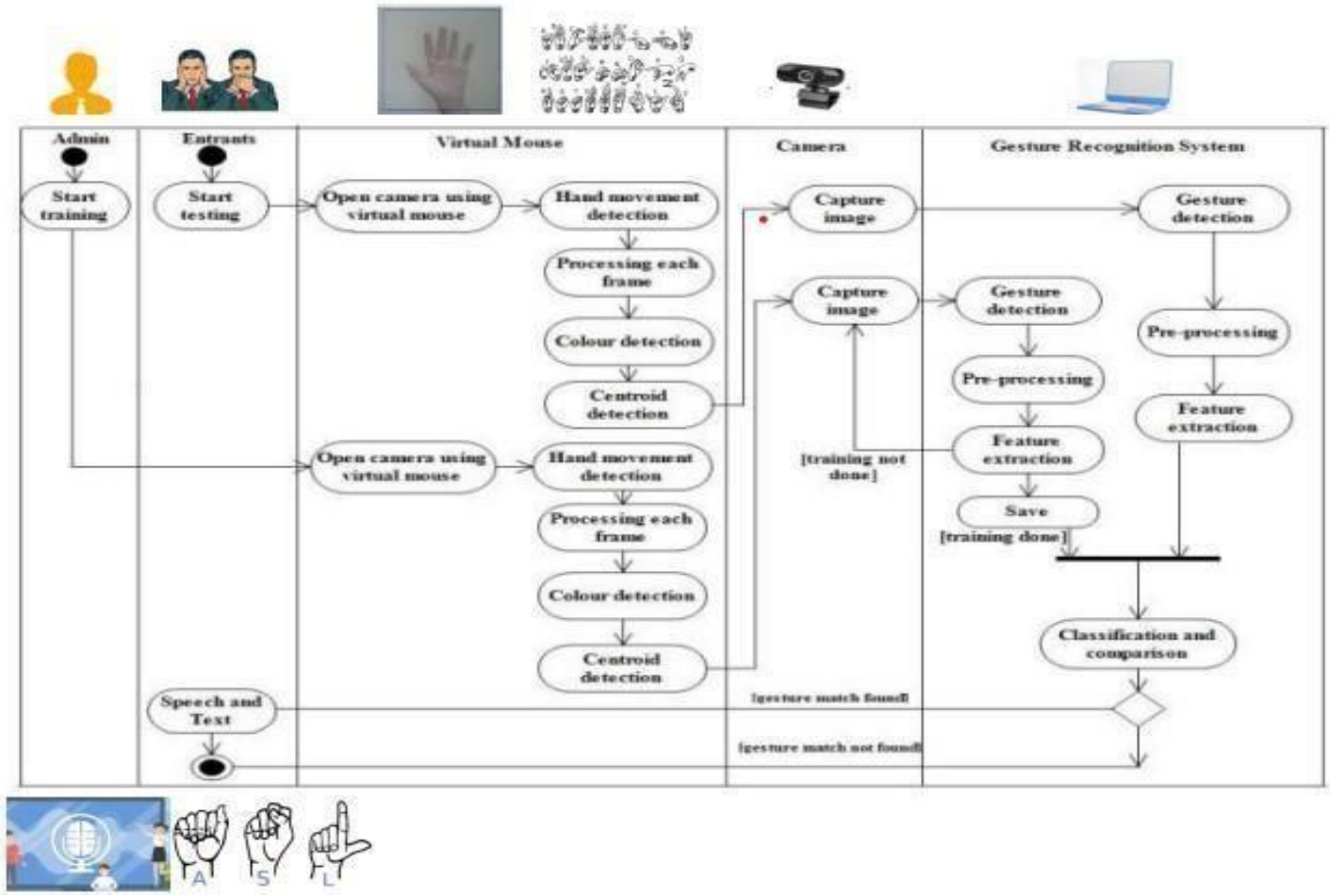
A data flow diagram (DFD) maps out the flow of information for any process or it uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

DFD for the given project is presented as shown below,

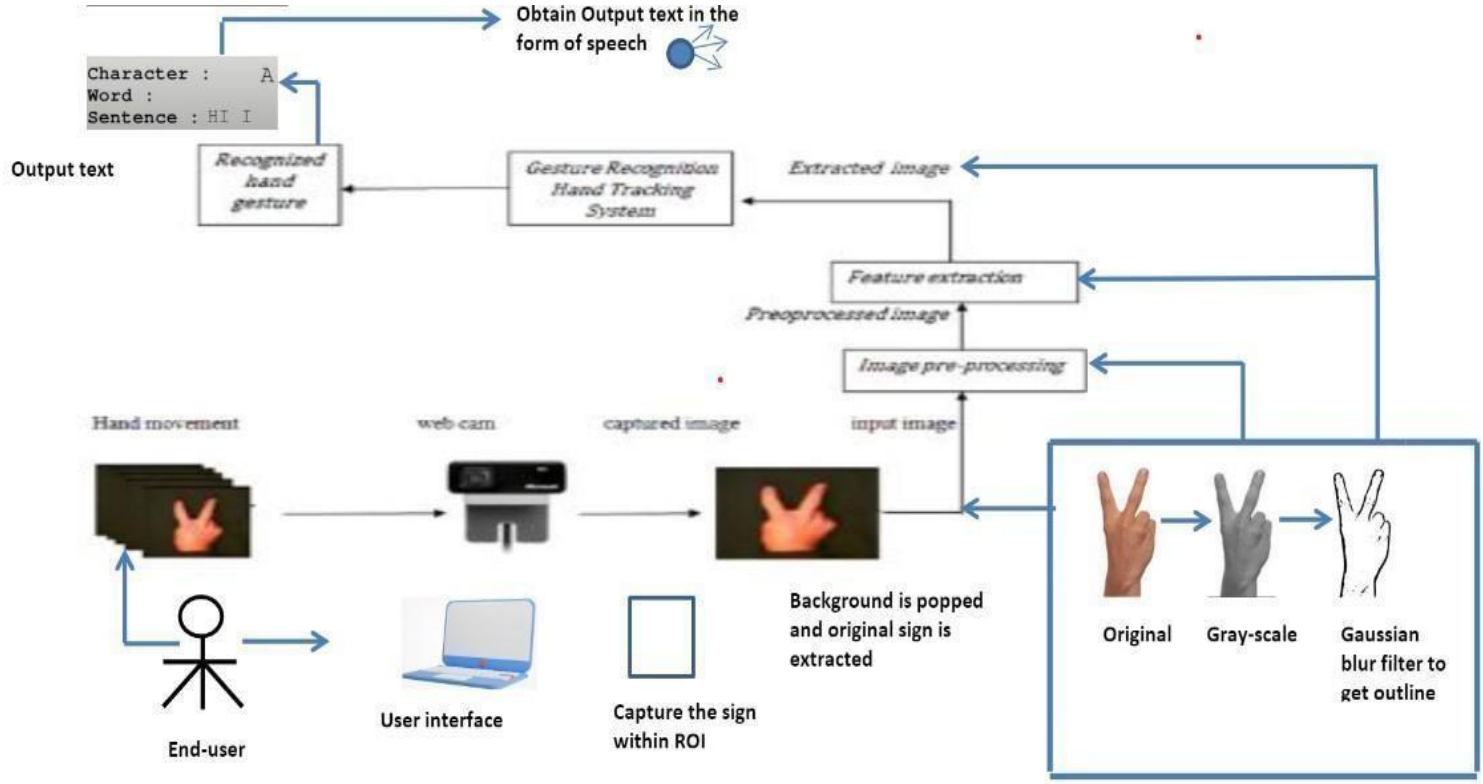


5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution architecture (SA) is an architectural description of a specific solution. The solution architecture **helps ensure that a new system will fit the existing enterprise environment**.



Technical Architecture (TA) involves development of technical blueprint with regard to the arrangement, interaction and interdependence of all elements so that system –relevant requirements are met.



5.3 USER STORIES

The step by step process involved in the usage of the app is mentioned in the tabular format as shown below.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer *(Desktop user)*	Registration	USN-1	**Not Required**	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	**Not Required**		High	Sprint-1
	Dashboard	USN-3	**Not Required**			
Customer *(Desktop user)*	**Main page**	USN-4	As a User, **I can enter the web page once clicked, which provides be the Guidelines** to use the app	I can **enter the web page** once clicked	Medium	Sprint-1
Customer (Desktop user)	**Guidelines**	USN-5	As a User , **I can give a read through the guidelines to understand the functioning of the app.**	I can **give a read through the guidelines.**	Medium	Sprint-1
Customer *(Desktop user)*	**Convert Sign**	USN-6	As a User, **I can click the button Convert sign** , which directs me towards the Main screen	I can **click the button Convert sign and directed me to main screen.**	Medium	Sprint-2
Customer *(Desktop user)*	**Camera(Hand movement detection)**	USN-7	As a User, **I can show my hand sign towards the camera which converts them into text manner.**	**I can show my hand sign** towards the camera accurately.	High	Sprint-2
Customer *(Desktop user)*	**Voice mode**	USN-8	Once the text is obtained, As a User **I can click on the voice mode which provides the text in the form of speech.**	**I can click on the voice mode** which provides the text in the form of speech.	High	Sprint-2
Customer Care Executive	**Provide the necessary functionalities required to use the app.**		As an Executive, I can **provide the Specifications of Camera required, and other factors** that are required for smooth functioning of the app.	**I can provide the Specifications** of Camera required, and other factors	Low	Sprint-1
Customer Care Executive	**Check the performance of the app**		As an Executive, I can **check the usage and queries obtained from the end users**.	I can **check the usage and queries** obtained from the end users.	Medium	Sprint-1
Administrator	**Receive queries based on the usage**		As an Admin, I can **take the queries from the customer care and perform the testing phase again , loading the other signs in the dataset**, in order to make the customers to use the app effectively.	I can **take the queries from the customer care and perform necessary phases again.**	High	Sprint-2

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	Not Required	2	High	-
Sprint-1	Login	USN-2	Not Required	1	High	-
Sprint-1	Main page	USN-4	As a User, **I can enter the web page once clicked, which provides be the Guidelines** to use the app	1	Medium	**NAGENDRA PRASANTH A** **SAI VIGNESH S**
Sprint-1	Guidelines	USN-5	As a User , **I can give a read through the guidelines to understand the functioning of the app.**	1	Medium	**NAGENDRA PRASANTH A** **SAI VIGNESH S** **SIVA BALAJI P**
Sprint-2	Convert Sign	USN-6	As a User, **I can click the button Convert sign** , which directs me towards the Main screen	4	Medium	**NAGENDRA PRASANTH A** **SAI VIGNESH S** **SIVA BALAJI P** **KRISHNA RAJ R**

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	**Camera(Hand movement detection)**	USN-7	As a User, **I can show my hand sign towards the camera which converts them into text manner.**	8	High	**NAGENDRA PRASANTH A** **SAI VIGNESH S**
Sprint-3	Voice mode	USN-8	Once the text is obtained, As a User **I can click on the voice mode which provides the text in the form of speech.**	3	High	**NAGENDRA PRASANTH A** **SAI VIGNESH S** **SIVA BALAJI P**
Sprint-1	**Provide the necessary functionalities required to use the app.**		As an Executive, I can **provide the Specifications of Camera required, and other factors** that are required for smooth functioning of the app.	1	Low	**NAGENDRA PRASANTH A** **SAI VIGNESH S**
Sprint-4	**Check the performance of the app**		As an Executive, I can **check the usage and queries obtained from the end users**.	1	Medium	**NAGENDRA PRASANTH A** **SIVA BALAJI P**
Sprint-4	**Receive queries based on the usage**		As an Admin, I can **take the queries from the customer care and perform the testing phase again , loading the other signs in the dataset**, in order to make the customers to use the app effectively.	2	High	**NAGENDRA PRASANTH A** **SIVA BALAJI P** **SAI VIGNESH S** **ANANDARAJ K**

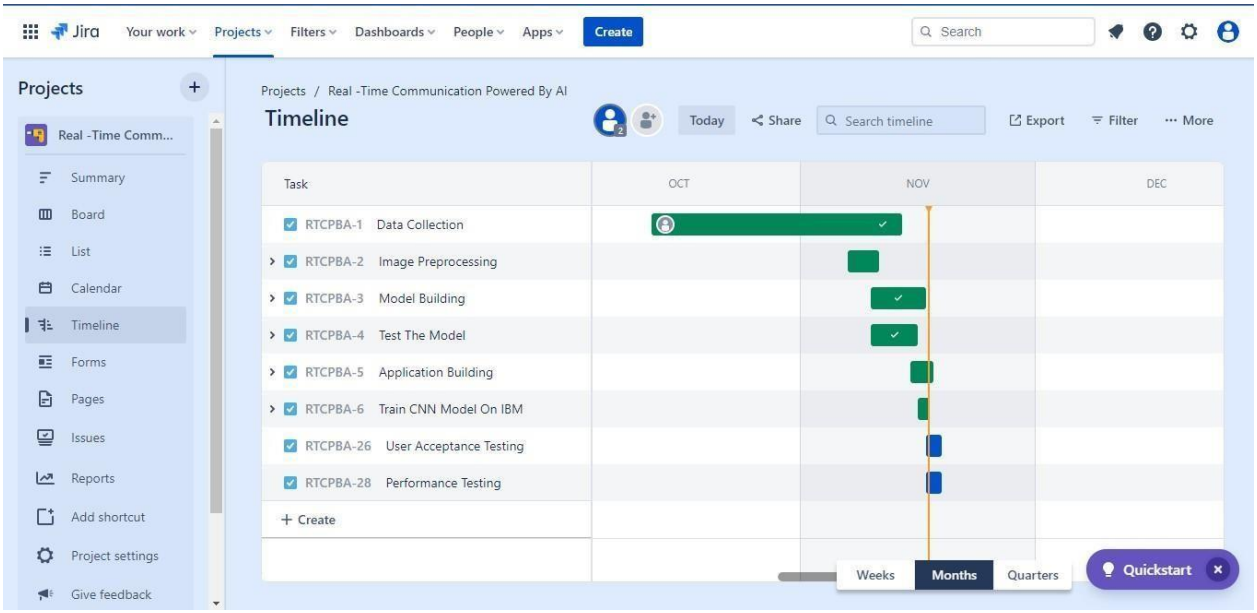
6.2 SPRINT DELIVERY SCHEDUL

The estimated time of delivery required for each sprint is mentioned clearly in the following tabular column.

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	6	6 Days	06 Nov 2022	12 Nov 2022		12 Nov 2022
Sprint-2	12	12 Days	01 Nov 2022	12 Nov 2022		12 Nov 2022
Sprint-3	3	3 Days	13 Nov 2022	15 Nov 2022		15 Nov 2022
Sprint-4	4	4 Days	17 Nov 2022	20 Nov 2022		20 Nov 2022

6.3 REPORTS FROM JIRA

JIRA Software is part of a family of products designed to help teams of all types manage work**.** It can be accepted as an task scheduler which describes the To-do, In-Progress and Done tasks.



CHAPTER 7
CODING & SOLUTIONING

7.1 FEATURE 1

Machine Learning cannot be extensively used for feature extraction because its algorithm cannot handle high dimensional data and also it is one of the big challenges in object detection, image classification, etc. Deep Learning inspired from how biological neural network have been playing a key role in the field of machine learning. As we are aware of the fact that images are high dimensional vectors. It would take a huge number of parameters to characterize the network, if we take flatten image feature vector 1-D. To address this problem (of taking long vector), CNN are proposed to reduce the number of parameters and adapting the network architecture specifically to vision tasks. CNN works well on tasks. Convolution Neural Network basically includes set of layers each having their own functionalities. CNN works by extracting features from images. There is no need for manual feature extraction. Features are trained while the network trains on the set of images.

CNNs key components includes:

Convolutional layer:

Consist of grouped neuron in a rectangular grid. It is the application of Filter to an input that results in an activation. Repeated activation of it results into a feature map, which indicating the locations and strength of a detected features in an input that is image.

Activation function:

Decides whether a neuron should fire an output or not and it is also used to increase the non-linearity into the output of the neuron.

Pooling layers :

Are present after a single or a set of convolution layers. Its function is to down sample the detection of features in feature map.

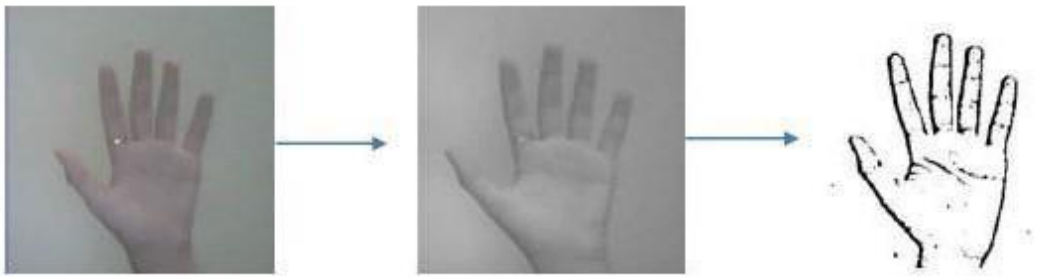
Fully connected layer:

Dense layer which is the final learning phase where classification takes place. Every node of it is connected to every nodes of previous layer.

Dataset Generation:

For this project we tried to make our own dataset for the ASL language but due to lack of resources we were unable to do so. Then we find out the already existing datasets that matched our requirements. All we could find were the datasets in the form of RGB values. Hence, we decided to transform it into our required form.

By using batch mode transformation/data augmentation we try to convert it into Gray scale Image as shown below:



Raw Image (RGB)

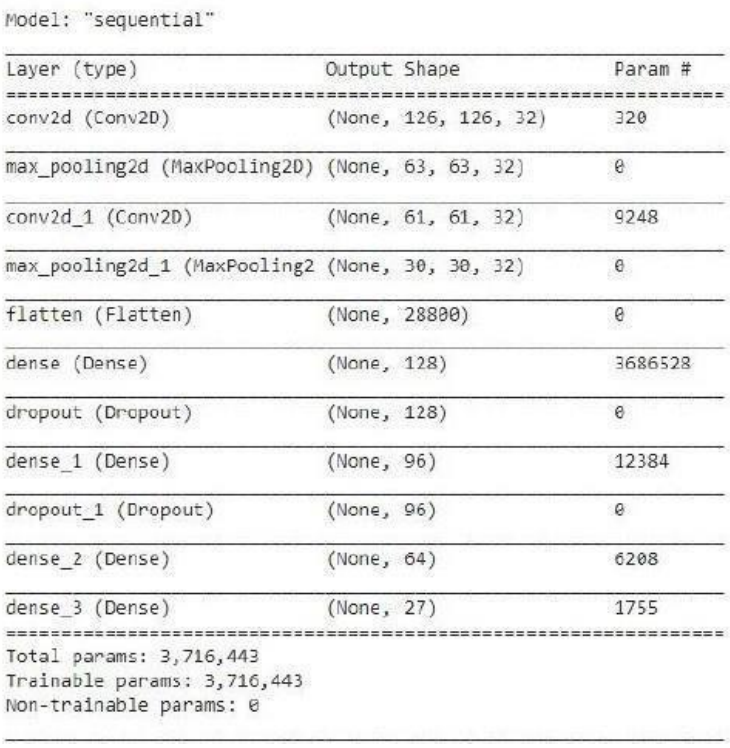
Gray Scale Image

Gaussian Blur

CNN architecture sum

It is one of the most important part for our network i.e. in defining our required architecture. A brief explanation of CNN is provided below along with a figure of our model architecture summary:

- Starts with an input image.
- Applies many different filters to obtain a feature map.
- Applies a RELU function to increase non-linearity.
- Applies pooling layer to each future map.
- Flattening the pooled images into one long vector.
- Inputs the vector into fully connected ANN.
- Dropout is also used to mitigate the overfitting.
- Final fully connected layer provides the voting of the classes.
- Trains through forward propagation and back propagation for many epochs



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 126, 126, 32)      320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 63, 63, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 61, 61, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 30, 30, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 28800)             0
_________________________________________________________________
dense (Dense)                (None, 128)               3686528
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 96)                12384
_________________________________________________________________
dropout_1 (Dropout)          (None, 96)                0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                6208
_________________________________________________________________
dense_3 (Dense)              (None, 27)                1755
=================================================================
Total params: 3,716,443
Trainable params: 3,716,443
Non-trainable params: 0
_________________________________________________________________
```

Training our network:

After successfully completing all the steps of constructing our network. Now the next step is to train our network. To train the model effectively we have set epochs and other necessary required configurations to train it effectively.

Saving model:

The metadata file (or model.json) in a Common Data Model folder describes the data in the folder, metadata and location, as well as how the file was generated and by which data producer. Metadata summarizes basic information about data, which can make finding and working with particular instances of data easier. JSON (JavaScript Object Notation) is a popular data format used for representing structured data. So, keeping in the mind regarding usability of data model we have saved our model as json file.

Gesture Classification:

Our approach uses two layers of algorithm to predict the final symbol of the user.

Algorithm Layer 1:

- Apply gaussian blur filter and threshold to the frame taken with Open CV to go the processed image after feature extraction.
- This processed image is passed to the CNN model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.
- Space between the words is considered using the blank symbol.

Algorithm Layer 2:

- We detect various sets of symbols which show similar results on getting detected.
- We then classify between those sets using classifiers made for those sets only.

Activation Layer :

We have used ReLu (Rectified Linear Unit) in each of the layers (convolutional as well as fully connected neurons). ReLu calculates max(x,0) for each input pixel. This adds nonlinearity to the formula and helps to learn more complicated features. It helps in removing the vanishing gradient problem and speeding up the training by reducing the computation time.

Pooling Layer :

We apply Max pooling to the input image with a pool size of (2, 2) with relu activation function. This reduces the amount of parameters thus lessening the computation cost and reduces overfitting.

Dropout Layer:

The problem of overfitting, where after training, the weights of the network are so tuned to the training examples they are given that the network doesn't perform well when given new examples. This layer "drops out" a random set of activations in that layer by setting them to zero. The network should be able to provide the right classification or output for a specific example even if some of the activations are dropped out.

Considering the above features , the following main application code is generated.

```
from PyQt5.QtWidgets import QApplication
from PyQt5.QtWidgets import QWidget

from PyQt5.QtGui import QImage
from PyQt5.QtGui import QPixmap
from PyQt5.QtCore import QTimer


from PyQt5 import QtCore, QtGui, QtWidgets

import tensorflow.keras as tf
import cv2
import numpy as np
import operator
import sys, os
import enchant
from string import ascii_uppercase
import turtle
import time
from multiprocessing import Process
from PyQt5 import QtCore, QtGui, QtWidgets
import pyttsx3
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QMainWindow, QAction, qApp, QApplication


engine = pyttsx3.init()
def loader():

turtle.setup(900,  500)
t = turtle.Turtle()
screen = turtle.Screen()
screen.title("Sign To Text")
screen.bgpic("t.png")
t.hideturtle()

t.penup()
t.goto(0, 110)
t.pendown()
t.pencolor("black")
t.write("Sign To Text", move='true', align="center", font=("Georgia", 50))
t.penup()

t.goto(-20, 80)
t.pendown()
t.write("For Deaf & Dumbs", align="left", font=("georgia", 15, "normal",
    'italic'))
t.penup()
t.goto(-20, 60)
t.pendown()

t.penup()
t.goto(-207, -130)
t.pendown()
t.pencolor("white")
```

```
    t.write("Initializing .......", align='right', font=("Ariel", 13, 'normal',
'italic'))



    t.pensize(3)
    t.penup()
    t.goto(-300, -100)
    t.pendown()

    t.forward(200)
    time.sleep(9)
    t.forward(150)
    time.sleep(9)
    t.forward(200)

    time.sleep(9)
    t.forward(100)
    time.sleep(9)
    t.forward(90)
    time.sleep(9)
    turtle.bye()

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(801, 626)
        palette = QtGui.QPalette()
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Light, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Midlight, brush)
        brush = QtGui.QBrush(QtGui.QColor(127, 127, 127))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Dark, brush)
        brush = QtGui.QBrush(QtGui.QColor(170, 170, 170))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Mid, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.BrightText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ButtonText, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)

        brush.setStyle(QtCore.Qt.SolidPattern)
```

```
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Light, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Midlight, brush)
brush = QtGui.QBrush(QtGui.QColor(127, 127, 127))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Dark, brush)
brush = QtGui.QBrush(QtGui.QColor(170, 170, 170))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Mid, brush)
brush = QtGui.QBrush(QtGui.QColor(127, 127, 127))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Text, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.BrightText, brush)
brush = QtGui.QBrush(QtGui.QColor(127, 127, 127))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ButtonText, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled,  QtGui.QPalette.Base,  brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Window, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Shadow, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.AlternateBase, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipBase, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipText, brush)
MainWindow.setPalette(palette)
font = QtGui.QFont()
font.setPointSize(8)
MainWindow.setFont(font)
MainWindow.setAutoFillBackground(True)
self.centralwidget = QtWidgets.QWidget(MainWindow)
self.centralwidget.setObjectName("centralwidget")
self.Heading = QtWidgets.QLabel(self.centralwidget)
self.Heading.setGeometry(QtCore.QRect(0, 0, 801, 41))
palette = QtGui.QPalette()
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
brush = QtGui.QBrush(QtGui.QColor(85, 170, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
brush = QtGui.QBrush(QtGui.QColor(213, 234, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Light, brush)
brush = QtGui.QBrush(QtGui.QColor(149, 202, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
brush.setStyle(QtCore.Qt.SolidPattern)
```

```
palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Base, brush)
brush = QtGui.QBrush(QtGui.QColor(85, 170, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Window, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Shadow, brush)
brush = QtGui.QBrush(QtGui.QColor(170, 212, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.AlternateBase, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ToolTipBase, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ToolTipText, brush)
brush = QtGui.QBrush(QtGui.QColor(42, 85, 127))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.WindowText, brush)
brush = QtGui.QBrush(QtGui.QColor(85, 170, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)
brush = QtGui.QBrush(QtGui.QColor(213, 234, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Light, brush)
brush = QtGui.QBrush(QtGui.QColor(149, 202, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Midlight, brush)
brush = QtGui.QBrush(QtGui.QColor(42, 85, 127))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Dark, brush)
brush = QtGui.QBrush(QtGui.QColor(56, 113, 170))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Mid, brush)
brush = QtGui.QBrush(QtGui.QColor(42, 85, 127))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Text, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.BrightText, brush)
brush = QtGui.QBrush(QtGui.QColor(42, 85, 127))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ButtonText, brush)
brush = QtGui.QBrush(QtGui.QColor(85, 170, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled,  QtGui.QPalette.Base,  brush)
brush = QtGui.QBrush(QtGui.QColor(85, 170, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Window, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Shadow, brush)
brush = QtGui.QBrush(QtGui.QColor(85, 170, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.AlternateBase, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipBase, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipText, brush)
self.Heading.setPalette(palette)
```

```
font = QtGui.QFont()
font.setFamily("Baskerville Old Face")
self.Heading.setFont(font)
self.Heading.setAutoFillBackground(True)
self.Heading.setObjectName("Heading")
self.Img = QtWidgets.QLabel(self.centralwidget)
self.Img.setGeometry(QtCore.QRect(400, 70, 381, 281))
self.Img.setText("")
self.Img.setPixmap(QtGui.QPixmap("signs.png"))
self.Img.setScaledContents(True)
self.Img.setObjectName("Img")
self.character = QtWidgets.QLabel(self.centralwidget)
self.character.setGeometry(QtCore.QRect(20, 340, 101, 31))
self.character.setObjectName("character")
self.word = QtWidgets.QLabel(self.centralwidget)
self.word.setGeometry(QtCore.QRect(10, 380, 101, 31))
self.word.setObjectName("word")
self.sentence = QtWidgets.QLabel(self.centralwidget)
self.sentence.setGeometry(QtCore.QRect(20, 420, 101, 31))
self.sentence.setObjectName("sentence")
self.ch = QtWidgets.QLabel(self.centralwidget)
self.ch.setGeometry(QtCore.QRect(130, 340, 101, 31))
palette = QtGui.QPalette()
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Light, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Midlight, brush)
brush = QtGui.QBrush(QtGui.QColor(127, 127, 127))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Dark, brush)
brush = QtGui.QBrush(QtGui.QColor(170, 170, 170))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Mid, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.BrightText, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ButtonText, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Window, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Shadow, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
self.sentence1.setObjectName("sentence1")
self.label = QtWidgets.QLabel(self.centralwidget)
```

```
self.label.setGeometry(QtCore.QRect(0, 460, 801, 31))
palette = QtGui.QPalette()
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
brush = QtGui.QBrush(QtGui.QColor(51, 200, 151))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
brush = QtGui.QBrush(QtGui.QColor(110, 255, 208))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Light, brush)
brush = QtGui.QBrush(QtGui.QColor(80, 227, 179))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Midlight, brush)
brush = QtGui.QBrush(QtGui.QColor(25, 100, 75))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Dark, brush)
brush = QtGui.QBrush(QtGui.QColor(34, 133, 101))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Mid, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.BrightText, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ButtonText, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)
brush = QtGui.QBrush(QtGui.QColor(51, 200, 151))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Window, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Shadow, brush)
brush = QtGui.QBrush(QtGui.QColor(153, 227, 203))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.AlternateBase, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ToolTipBase, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ToolTipText, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.WindowText, brush)
brush = QtGui.QBrush(QtGui.QColor(51, 200, 151))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Button, brush)
brush = QtGui.QBrush(QtGui.QColor(110, 255, 208))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Inactive,  QtGui.QPalette.Light,  brush)
brush = QtGui.QBrush(QtGui.QColor(80, 227, 179))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Midlight, brush)
brush = QtGui.QBrush(QtGui.QColor(25, 100, 75))
```

```
 brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Dark, brush)



def action1(self):
        self.hs.check(self.word)
        predicts = self.hs.suggest(self.word)
        if (len(predicts) > 0):
            self.word = ""
            self.str += " "
            self.str += predicts[0]

    def action2(self):
        self.hs.check(self.word)
        predicts = self.hs.suggest(self.word)
        if (len(predicts) > 1):
            self.word = ""
            self.str += " "
            self.str += predicts[1]


    def action3(self):
        self.hs.check(self.word)
        predicts = self.hs.suggest(self.word)
        if (len(predicts) > 2):
            self.word = ""
            self.str += " "
            self.str += predicts[2]

    def action4(self):
        self.hs.check(self.word)



        predicts = self.hs.suggest(self.word)
        if (len(predicts) > 3):
            self.word = ""
            self.str += " "
            self.str += predicts[3]

    def destructor(self):
        print("Closing Application...")
        root.destroy()
        self.vs.release()
        cv2.destroyAllWindows()

    def action5(self):
        self.str += " "
        self.str += self.word
        self.word = ""

    def delet(self):
        self.str = self.str.split()[:-1]
        self.str = ' '.join(self.str)

    def delw(self):
        self.word = self.word[:-1]

    def speak(self):
        try:
```

```
            engine.say(self.str)
            engine.runAndWait()
        except:
            print('No Word')


    print("Starting Application...")

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    loader()
    MainWindow.show()
    #ui.vs.release()
    sys.exit(app.exec_())
```

7.2 FEATURE 2

Implementation:

- Whenever the count of a letter detected exceeds a specific value and no other letter is close to it by a threshold, we print the letter and add it to the current string.
- Otherwise, we clear the current dictionary which has the count of detections of present symbol to avoid the probability of a wrong letter getting predicted.
- Whenever the count of a blank (plain background) detected exceeds a specific value and if the current buffer is empty no spaces are detected.
- In other case it predicts the end of word by printing a space and the current gets appended to the sentence below.

Autocorrective Feature:

A python library Hunspell_suggest is used to suggest correct alternatives for each (incorrect) input word and we display a set of words matching the current word in which the user can select a word to append it to the current sentence. This helps in reducing mistakes committed in spellings and assists in predicting complex words.

Training and Testing:

We convert our input images (RGB) into grayscale and apply gaussian blur to remove unnecessary noise. Then we have applied adaptive threshold to extract our hand from the background and resize our images to 128 x 128. We feed the input images after pre-processing to our model for training and testing after applying all the operations mentioned above. The prediction layer estimates how likely the image will fall under one of the classes. So, the output is normalized between 0 and 1 and such that the sum of each values in each class equals to 1. We have achieved this using SoftMax function..

User Interface:

UI stands for user interface and is the point of communication between human and the computer. The function of our UI is to provide users with an intuitive interaction and support that manages to provide a solution for the desired task. We have provided user input control's which are the interactive component of interface. It basically includes space for input gesture, various other spaces for character, word, sentence and buttons for audio output, backspace, reset and also for the suggestions to be picked up from the screen. We are providing both text and audio output on the same platform which makes a user to understand the input gestures more effectively.

As you can see the UI window below. At the top there is a title shown and exactly below it there is a space provided for gestures to displayed. And the on the right-hand side of it three buttons are given with the distinctive functionality which are as follow i.e., pressing the audio button one can have an audio output; backspace button to correct the wrong word being predicted; reset button to have a blank screen for new gestures to be classified. After that downside in the UI screen there are spaces given for result to be obtained i.e., for character, word, sentence. And below it few more spaces are provided for the suggestions to presented before a use to help me picking up the correct word in order to communicate effectively.

CHAPTER 8

TESTING

8.1 TEST CASES

For the given project , the following test cases involved are,

Main page	USN-4	As a User, **I can enter the web page once clicked, which provides be the Guidelines** to use the app	I can **enter the web page** once clicked
Guidelines	USN-5	As a User , **I can give a read through the guidelines to understand the functioning of the app.**	I can **give a read through the guidelines.**
Convert Sign	USN-6	As a User, **I can click the button Convert sign** , which directs me towards the Main screen	I can **click the button Convert sign and directed me to main screen.**
Camera(Hand movement detection)	USN-7	As a User, **I can show my hand sign towards the camera which converts them into text manner.**	**I can show my hand sign** towards the camera accurately.
Voice mode	USN-8	Once the text is obtained, As a User **I can click on the voice mode which provides the text in the form of speech.**	**I can click on the voice mode** which provides the text in the form of speech.
Provide the necessary functionalities required to use the app.		As an Executive, I can **provide the Specifications of Camera required, and other factors** that are required for smooth functioning of the app.	**I can provide the Specifications** of Camera required, and other factors
Check the performance of the app		As an Executive, I can **check the usage and queries obtained from the end users**.	I can **check the usage and queries** obtained from the end users.
Receive queries based on the usage		As an Admin, I can **take the queries from the customer care and perform the testing phase again , loading the other signs in the dataset**, in order to make the customers to use the app effectively.	I can **take the queries from the customer care and perform necessary phases again.**

8.2 USER ACCEPTANCE TESTING

Based on the following test cases, the user scenario in every test case is described in the below for as user acceptance testing.

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

Evaluating the performance of a CNN model is one of the important steps while building an effective CNN model. To evaluate the performance or quality of the model, different metrics are used, and these metrics are known as performance metrics or evaluation metrics. These performance metrics help us understand how well our model has performed for the given data. In this way, we can improve the model's performance by tuning the hyper-parameters. Each CNN model aims to generalize well on unseen/new data, and performance metrics help determine how well the model generalizes on the new dataset.

In a classification problem, the category or classes of data is identified based on training data. The model learns from the given dataset and then classifies the new data into classes or groups based on the training. It predicts class labels as the output, such as *Yes or No, 0 or 1, Spam or Not Spam,* etc. To evaluate the performance of a classification model, different metrics are used, and some of them are as follows:

- **Accuracy**
- **Confusion Matrix**
- **Precision**
- **Recall**
- **F-Score**
- **AUC(Area Under the Curve)-ROC**

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES :

- **REAL TIME FUNCTIONING**

The output of the sign language will be displayed in the text form in real time. This makes the system more efficient and hence communication of the hearing and speech impaired people more easy. The images captured through web cam are compared and the result of comparison is displayed at the same time. Thus this feature of the system makes communication very simple and delay free.

- **PORTABLE**

When this entire project is implemented on Raspberry Pie computer, which is very small yet powerful computer, the entire system becomes portable and can be taken anywhere. This feature facilitates the user to take the system anywhere and everywhere and overcomes the barrier of restricting him/herself to communicate without a desktop or laptop.

- **NO NEED OF CALIBRATION.**

DISADVANTAGES:

- Sign language requires the use of hands to make gestures. This can be a problem for people who do not have full use of their hands. Even seemingly manageable disabilities such as Parkinson's or arthritis can be a major problem for people who must communicate using sign language.
- Cost and is difficult to be used commercially

CHAPTER 11

CONCLUSION

Communication between deaf-mute and a normal person have always been a challenging task. The goal of our project is to reduce the barrier between them. We have made our effort by contributing to the field of Indian Sign Language recognition. In this project, we developed a CNN-based human hand gesture recognition system. The salient feature of our system is that there is no need to build a model for every gesture using hand features such as fingertips and contours. Here in this project, we have constructed a CNN classifier which is capable of recognizing sign language gestures. The proposed system has shown satisfactory results on the transitive gestures.

In this report, a functional real time vision-based sign language recognition for deaf and dumb people have been developed. We achieved final accuracy of 98.0% on our dataset. We are able to improve our prediction after implementing two layers of algorithms, we have also verified our result for the similar looking gesture which were more prone to misclassification. This way we are able to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate

CHAPTER 12

FUTURE SCOPE

We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms. We are also thinking of improving the pre-processing to predict gestures in low light conditions with a higher accuracy.

CHAPTER 13

APPENDIX

SOURCE CODE:

Home.html

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome</title>
<style>
body{
    background-image: url("1.webp");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
}
h1{
  font-family:Calibri;
  background-image: linear-gradient(to left,
  #553c9a, #00FF00);
}
p.round2 {
 border: #00FF00;
 border-radius: 8px;
 padding: 5px;
}
h2{
  font-family:comic sans ms;
}
li,p{
```

```
   font-family:comic sans ms;
   font-size:30px;
}
img {
  border-radius: 30px;
}
</style>
<body>
<b><center><h1 style="font-family:Calibri">Real-
Time Communication Powered By
AI</h1></center></b>
<marquee direction="left"
height="100px"><b><h2>When everyone else says you
can't , determination says <strong><u>YES YOU CAN
:)</u></strong></h2></b></marquee>
<img align="right"
src="file:///C:/Users/Admin/Downloads/Hand%20Sign
%20Detection/6.png" alt="">
<p>Want to try? ... <img
src="file:///C:/Users/Admin/Downloads/Hand%20Sign
%20Detection/question-mark-face-png-
11552248123qpvsndmve9.png" width=30px> Check it
out. </p>
<ul>
<li><a href="About Us.html"style="text-
decoration:none;">About us</a></li>
<li><a href="Use the app.html"style="text-
decoration:none;">How to use the app?</a></li>
<li><a href="Contact.html"style="text-
decoration:none;" >Contact us</a></li>
</ul>
```

```
</body>
</head>
</html>
```

About us.html

```
<!DOCTYPE html>
<html>
<head>
<title>About Us</title>
<style>
body{
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
}
h1,p{
   font-family:comic sans ms;
}
p3{
  font-family:comic sans ms;


font-size:30px;
}
.column   {
  float: left;
  width: 13.3%;
  padding: 40px;
}
```

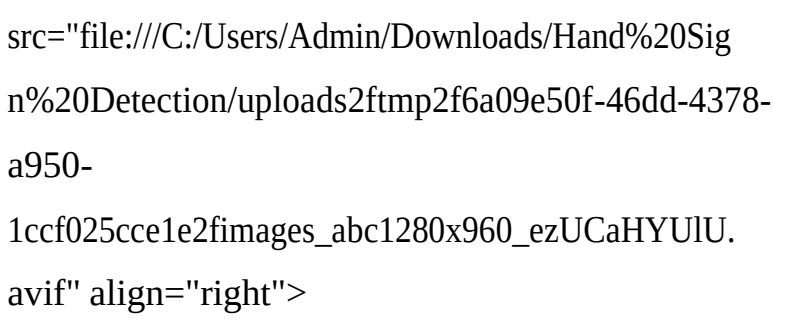
```
</style>
<body>
<h1><center><u>About     Us</u></center></h1>
<center><img
    src="file:///C:/Users/Admin/Downloads/download-
    removebg-preview.png" alt=" "></center>
<hr>
<p> The Project i.e. <p1 style="color:Green">Real
    Time Communication Powered By AI</p1> was
    made by the students of PCET in the view of
    <p2 style="color:Green"> Specially Disabled people
    </p2> who find extreme difficult , involving in the
    day to day communication and accepting negative
    comments from the outsiders.
<p>Hope this project will satisfy the requirements of the
    end users and motivate them in all walks of life</p>
<hr>

<p>To know about the Project in detail,<a
    href="https://github.com/IBM-EPBL/IBM-Project-
    48531-1660808738"style="text-
    decoration:none;">Click here</a></p>
<pre>

<p3><u>Contributed  by,</u></p3>
 <div class="row">
  <div class="column">
    <img
    src="file:///C:/Users/Admin/Downloads/Hand%20Si
    gn%20Detection/2.jpeg" alt=""
    style="width:100px"><p>NAGENDRA<br
```

```
    >PRASANTH A</p>
  </div>
  <div class="column">
    <img
      src="file:///C:/Users/Admin/Downloads/Hand%20Si
      gn%20Detection/3.jpeg" alt=""
      style="width:100px"><p>KRISHNARAJ
      R</p>
  </div>
  <div class="column">
    <img
      src="file:///C:/Users/Admin/Downloads/Hand%20Si
      gn%20Detection/4.jpeg" alt=""
      style="width:100px"><p>SAI VIGNESH
      S</p>

</div>
  <div class="column">
    <img
      src="file:///C:/Users/Admin/Downloads/Hand%20Si
      gn%20Detection/5.jpeg" alt=""
      style="width:100px"><p>SIVA BALAJI P</p>
  </div>
  <img
    src="file:///C:/Users/Admin/Downloads/Hand%20Si
    gn%20Detection/5.jpeg" alt=""
    style="width:100px"><p>ANANDARAJ K</p>

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

Use the app.html

```
<!DOCTYPE html>
<html>
<head>
<title>How to use the app?</title>
<style>
body{
     background-image: url("1.webp");
     background-repeat: no-repeat;
     background-attachment: fixed;
     background-size: cover;
    }
    h1{
       font-family:comic sans ms;
       color: Green;
    }
    h2,p,li{
      font-family:comic sans ms;
    }
    img{
     border-radius: 30px;
    }
    .column   {
      float: left;
      width: 13.3%;
      padding: 10px;
    }
    </style>
    <body>
```

<h1> <u>How to use the app?</u> </h1>



Sign language recognition is still a challenging problem inspire of many research efforts during the last many years.One of the methods of hand gesture recognition is to use the hand gloves for human computer interaction. But this method is sophisticated as it requires user to wear glove and carry a load of cables connecting the device to a computer.<p1 style="color:Green">Therefore, to eliminate this complication and to make user interaction with computer easy and natural we proposed to work on sign recognition using bare hands i.e., no usage of any external wearable hardware</p1>

<hr>

<u>Steps to be followed:</u>

First of all, Click here to download the zip file.

On Successful downloading,Select the application file to navigate to the app.

```
        <li>Use the hand sign provided here and
face towards the camera for text conversion</li>
        <li>After the generation of text , select the
speaker button to generate the voice of the resultant
text.</li>
        </ul>
        <pec>

        <h2> Sample photos are being provided
</h2>
        <div class="row">
          <div class="column">
            <img
src="file:///C:/Users/Admin/Downloads/Hand%20Sig
n%20Detection/7.jpg" alt=""
style="width:100%"><p>Generate the text</p>
          </div>
          <div class="column">
            <img
src="file:///C:/Users/Admin/Downloads/Hand%20Sig
n%20Detection/8.jpg" alt=""
style="width:100%"><p>Click the speaker</p>
          </div>
```

Contact.html

```
<!DOCTYPE html>
<html>
<head>
<title>Contact Us</title>
<style>
body{
    background-image: url("1.webp");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
}
h1{
  font-family:comic sans ms;
  color: Green;
}
h2{
  font-family:comic sans ms;
}
img{
 border-radius:20px;
}
</style>
<body>
<center><h1><u>Contact Us</u></h1></center>
<hr>
```

<h2>If any issues faced, while using the app, Don't hesitate to fill this form below. We will reach out to you shortly</h2>

<h2>Click here</h2>

<img src="file:///C:/Users/Admin/Downloads/Hand%20Sign%20Detection/9.jpg" width="1000">

</body>

</html>

</head>

GITHUB LINK:

GitHub - IBM-EPBL_IBM-Project-50829-1660925642

PROJECT DEMO LINK:

https://www.youtube.com/playlist?list=PLZ4cMZRhzA6yU7oxv1LN2Q4od3hWWeVnU