Real-Time Communication System Powered by AI for Specially Abled

TeamID: **PNT2022TMID53097**

Team Members:

Raghavesh DB

Padmesh C

K V Partheban

Shiva G

Introduction

Project Overview

In our society, we have people with disabilities. The technology is developing day by day, but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult.

1. 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.

2. The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb.

3. 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

.

Purpose

There are close to 2.2 percent of the Indian population with some form of disability.

The specially abled people feel left out, isolated from the common world. They face a of difficulty in expressing their feeling and even a basic communication towards other people.

They can just communicate among themselves , and very very basic needs communicated to other people.

In case of emergencies, it becomes even more difficult for them to tell and for us to make sense of what that means.

This project mainly focusses on giving a hope for the specially abled to communicate freely with other without any hesitation .

There are many talents among the specially abled , who are not given their chances just because of lack of communication.

They sometimes of denied of their rights and treated awkwardly and they don’t even have a means to voice out against it.

So this project is like a boon to these people who face problems everyday. This is a stepping stone in the communication of specially abled people and if implemented successfully , it helps in bridging the gap between the specially abled and the others.

Literature Survey

An IoT based device to facilitate communication for the disabled people was explored by many researchers (Pritesh Ambavane et al, 2020)[1] The proposed sensor-based system stored values in a microcontroller which were then transferred wirelessly to a mobile device. These values were then compared with the data stored on the device to display the corresponding text, later converted into speech using a text-to-speech conversion tool.

A Dumb's Communication System that transforms sign language into audio was proposed (Lillian Al Tinnawi et al, 2017)[2] The device utilizes EMG electrodes as a non-invasive wearable technology that provides accurate data on muscle motion. The data is analyzed with the help of a software in a separate circuit, which in combination with a previous database, converts the gesture into a real-time audio voice, delivered via a speaker. The circuit incorporates a Text-to-Speech (TTS) conversion unit for interpreting the matched gestures.

A gesture-based device for the communication of a deaf and dumb person has been implemented (Pallavi Verma et al, 2013)[3] A pair of gloves with a flex sensor is used to capture the movement of the user and is converted into a voltage with the help of the voltage divider rule. The digitized data is transmitted and matched by a gesture recognition system at the receiver, which delivers the matched data as input to an audio device.

A system to facilitate communication between the disabled and the normal person was implemented by a group of researchers (Aditi Kalsh et al, 2013)[4] The system is based on static hand gesture, captured using web camera processing and stored as a DCR file. Edge detection is performed using the canny edge algorithm followed by the wavelet family method to recognize the tips of the fingers. The gestures are

then matched with a set stored on a database to produce the final output.

REFERENCES

[1] Ambavane, Pritesh & Karjavkar, Rahul & Pathare, Hemant & Relekar, Shubham & Alte, Bhavana & Sharma, Neeraj, “A Novel Communication System For Deaf And Dumb People using gesture”, ITM Web of Conferences, 2020, 32. 02003. 10.1051/itmconf/20203202003.

[2] Tinawi, Lillian & Harb, Reem & Nasser, Hassan-Roland & Zaylaa, Amira & Hamawy, Lara, “A New Dumb's Communication System”, 2017.

[3] Verma, Pallavi & Priyadarshani, Richa, “Design of Communication Interpreter for Deaf and Dumb Person”, International Journal of Science and Research, 2013, 4. 2640-2643.

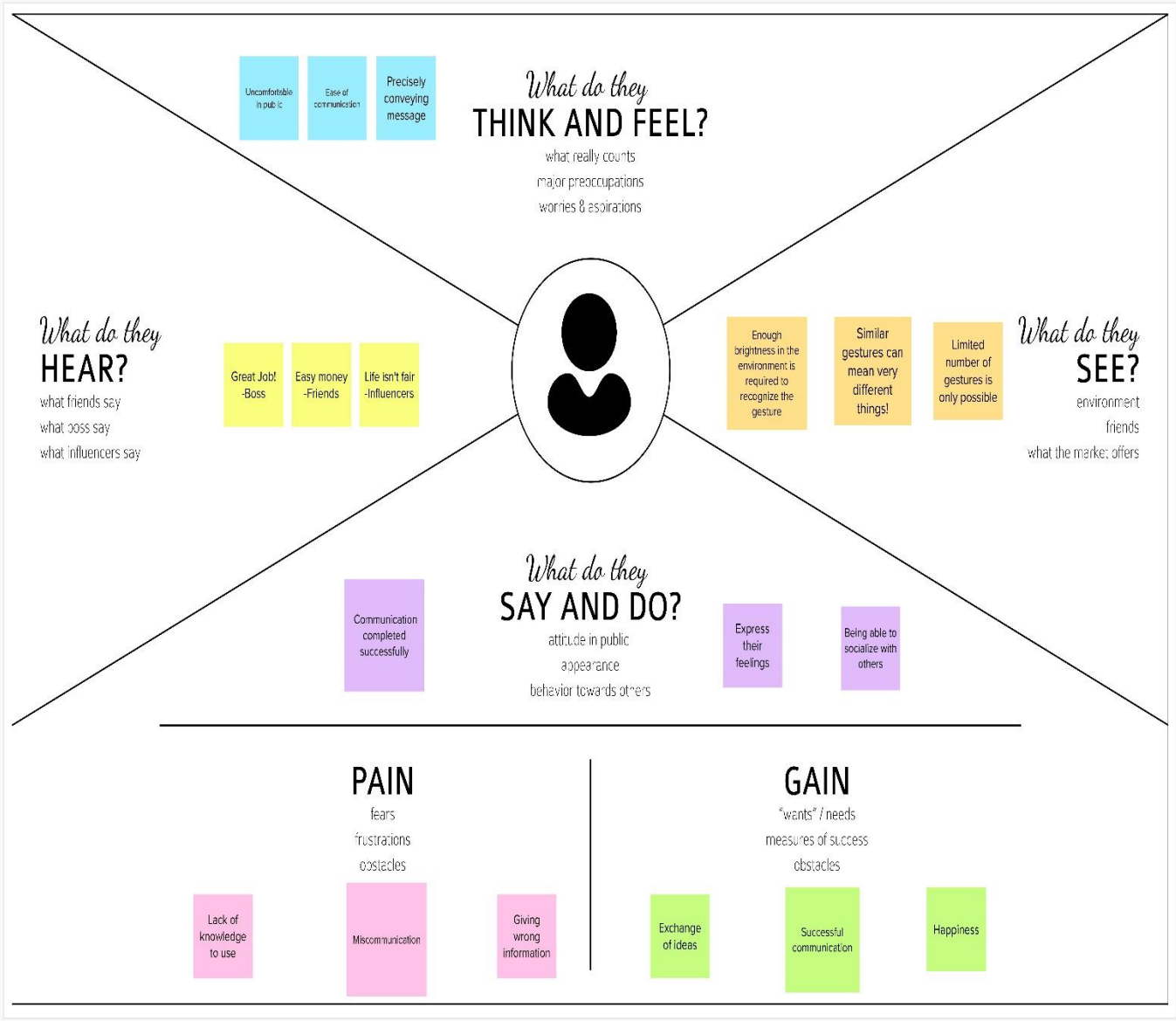
[4] Aditi Kalsh and N.S. Garewa, “Sign Language Rcognition System,”,International Journal of Computational Engineering Research, Vol 03, Issue 6, June 2013.

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

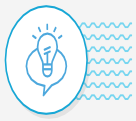
Build empathy and keep your focus on the user by putting yourself in their shoes.



Share your feedback

Brainstorming

Template



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

To Develop a Communication System which can enable faster communication between specially abled and normal people

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Share template feedback

Need some inspiration?

See a finished version of this template to

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Raghavesh

- Cheeks muscle movement can be used to word they are going to speak
- The Voice Assistant must be available in all local languages
- Voice assistants used for communication of Visually Challenged People
- Training the system with tons of data
- The system must be live all the time
- The System must work both online and offline

Partheban

- Design the application with inclusivity to all types of people.
- Research should based on data centric perspective
- Translation should be faultless
- An alarm button can be setup for any emergency uses for the specially abled person
- A notifcation to remind them about their daily tasks
- Modyfying goals based on circumstances

Padmesh

- Video of hand gestures converted into speech
- The system must be able to recognize braille language
- The Device must be small and are easily portable
- The Battery must last longer and can work even with an emergency backup
- The system can be operated by voice commands and buttons
- The user should feel like a normal person by using this device

Shiva

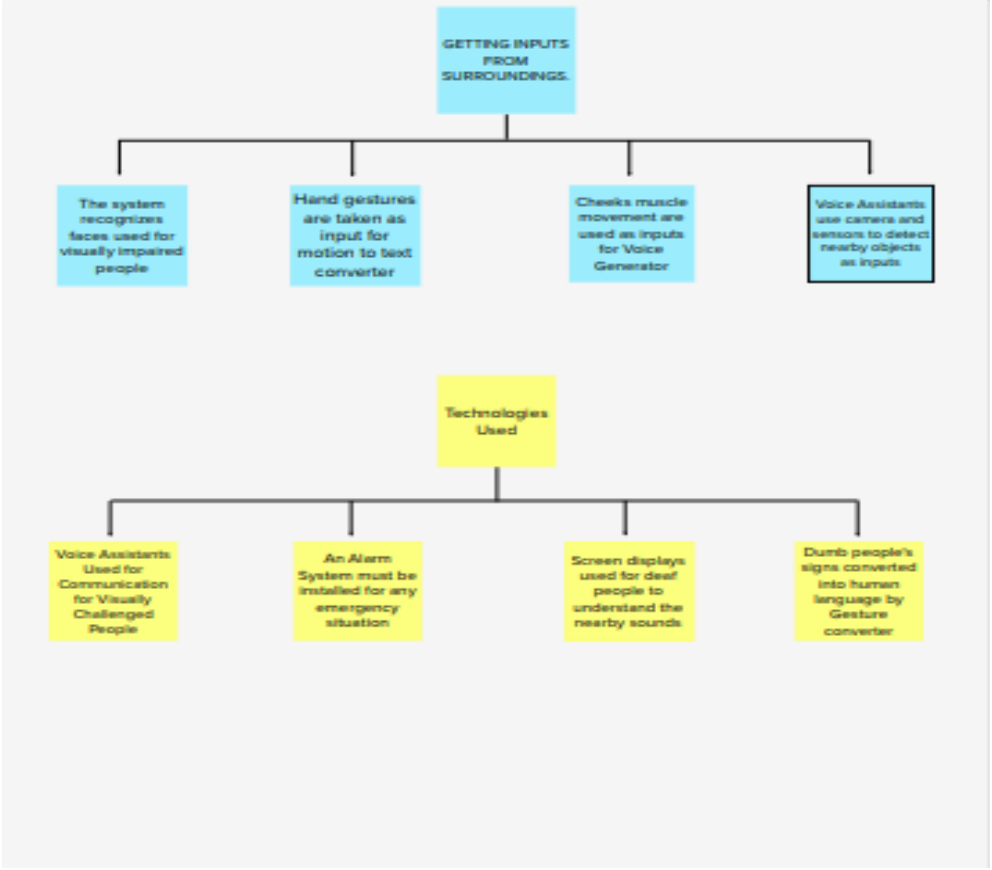
- The machine is made to learn all the actions so that it easily detects the movement of the person .
- Real-time captioning or translations for people with a hearing impairment or even people who don't speak the language.
- Text summarization for people with a mental impairment.
- Feedback from users for further improvement
- Facial recognition for people with a visual impairment.
- Inexpensive Translation Service

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.

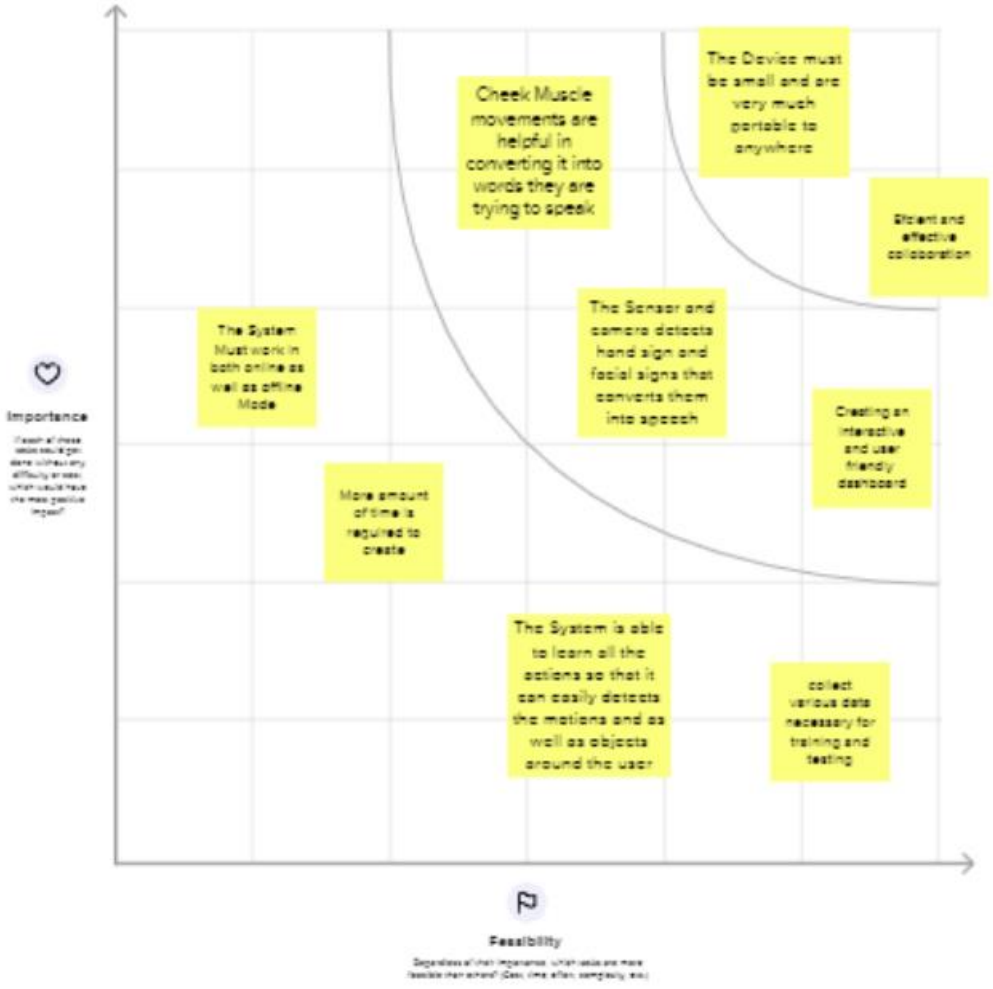
20 minutes



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



Proposed Solution Template

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

1. Problem Statement (Problem to be solved):

- Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people since normal people are not trained on hand sign language in emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Making communication between deaf-mute people and normal person effective is the problem to solve here.

2. Idea/Solution description:

- ❖ We aim to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

3. Novelty / Uniqueness:

a. Since deaf-mute people are usually deprived of normal communication with other people, they must rely on an interpreter or some visual communication. Now the interpreter cannot be available always, so this project can help eliminate the dependency on the interpreter.

b. A web-based version of the application will increase the reach to more people.

c. Integrating hand gesture recognition system using computer vision for establishing 2-way communication system.

4. Social Impact/ Customer Satisfaction:

❖ Sign language is a visual language that is used by deaf people as their mother tongue. Unlike acoustically conveyed sound patterns, sign language uses body language and manual communication to fluidly convey the thoughts of a person. It is achieved by simultaneously combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions. It can be used by a person who has difficulties in speaking or by a person who can hear but could not speak and by normal people to communicate with hearing disabled people. Therefore developing an intelligent system to recognize the sign language and convert it to voice will be massive aid for deaf-mute people to interact with the social world without any hassle and also since the machine takes care of the conversion the need for the normal people to have a good amount of knowledge will be eliminated.

5. Business Model (Revenue Model):

- The subscription model is an increasingly popular business model that involves a company selling a service via a subscription as opposed to a one-off product. If the service is going to be helpful as well as useful most customers will choose to auto-renew their subscriptions, which ensures that they always have the service provided to them. Because of recurring sales, this model allows you to predict what the yearly revenues are going to be, which is fantastic when trying to bring in new investors

6. Scalability of the Solution:

- As the technology advances the AI can improvise and more no of people can be incorporated with the system and the system can be extended to incorporate the knowledge of facial expressions and body language too so that there is a complete understanding of the context and tone of the input speech

Problem-Solution Fit

Problem-Solution fit canvas 2.0

Purpose/Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

Who is your customer?

1. Specially abled persons.
2. People who lost their speech or hearing ability by birth or due to some factors.
3. People with deaf and mute disabilities who struggles in communication.

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.

1. Low Budget, Proper Network Connections, Available devices for customer requirements
2. Difficult accessibility, not user friendly, more technical knowledge to handle.
3. There are.so many choices available but due to these constraints, choice of solutions were limited.

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

1. The first ever approach to sign language has only six sign gestures detection and using coloured hands for hand position recognition.
2. In this product, we provide feedback pop-up notifications frequently and an emergency purpose ping for people who have minimum knowledge about the application.

Explore AS, differentiate

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

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.

1. Deaf and Dumb people couldn't able to convey their messages easily to normal people.
2. Deaf people cannot hear the words as others speaks and they cannot express their feelings by words.
3. Concentrate on making their communication effective by means of concentrating more on listening and help them to live a normal life.

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.

1. This kind of disabilities also occur genetically.
2. It can also be caused due to accidents, injuries, obesity, infections or other illness.
3. The old methods use traditional translators which take too much of time to process.

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)

1. The customer should be provided with a customer care number and it gives many feedback pop-up notifications frequently which helps the customers to contact with us and get their jobs done.
2. In the device, there is an option named problem detection display in which can see the list of types of problem among them the customers should select their problem and the solution will be displayed for their respective problem.

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

1. Lack of communication with normal people will crack their mental strength
2. To advertise the product in specially abled schools and other places and creating awareness through social media.

4. EMOTIONS: BEFORE / AFTER — EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before using this product specially abled people were in struggle to communicate with other people.
But after using this product they are comfortable to communicate and led a usual life with happiness.

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 behaviour.

1. Voice recognition and predictive texting tools allow people to communicate easily using AI. We can also use AI sensors to monitor the health and save those data for future use.
2. In Particular, Using SSD ML algorithm recognizes the signs as words faster compared to old traditional translators. Because in old translators, every alphabet is to be recognized to form a whole statement.

8. CHANNELS of BEHAVIOUR — CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

Customers can use online voice assistants such as Siri, Google, ALexa to make use of their devices through online, Which help them to make use their app everywhere through online.

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Connecting with people might be difficult depending on the disabilities. Technology and AI leave no one behind and can benefit person with disability and able to learn sign language.

Extract online & offline CH of BE

Created by Daria Nepriakhina / Amaltama.com

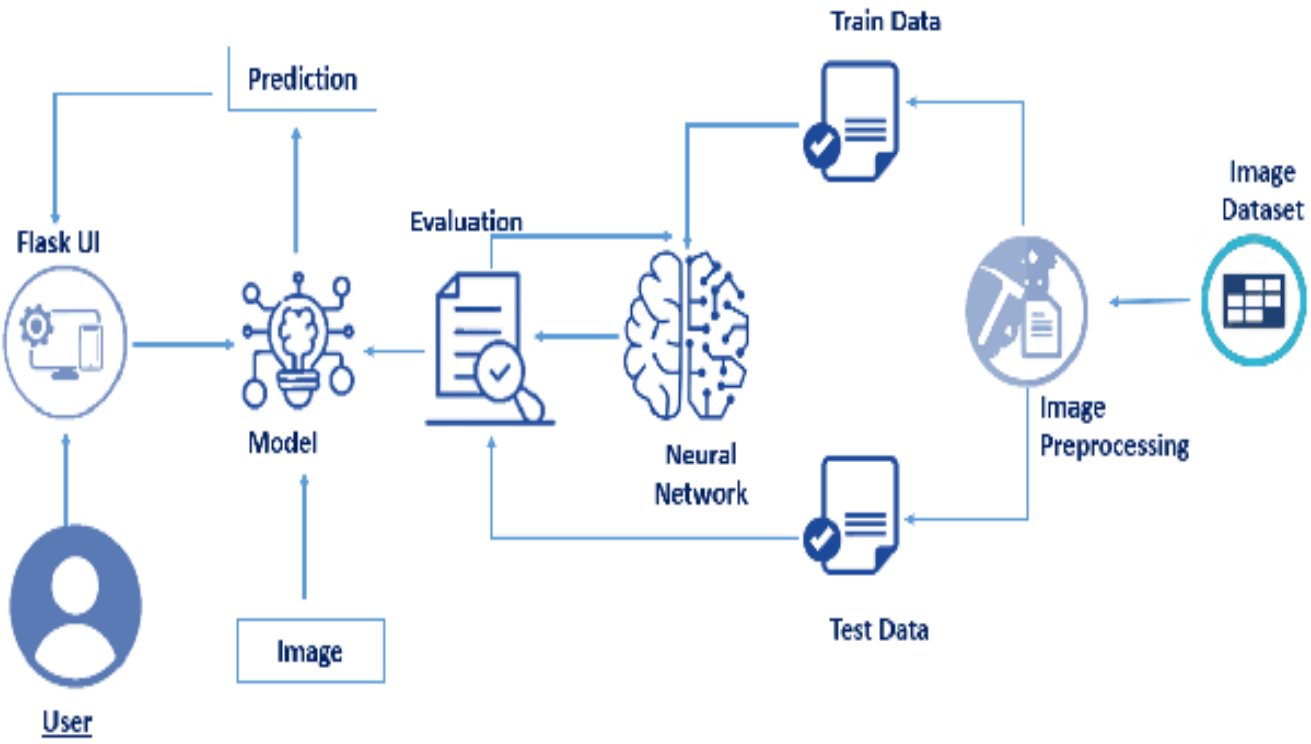
AMALTAMA

Solution Architecture

Solution architecture:

- It 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.

Example - Solution Architecture Diagram:



Project Design Phase-II
Solution Requirements

❖ Functional Requirements:

- Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Verification	The user should receive a verification e-mail which they have to confirm to complete the registration.
FR-4	Authentication	The data inside the device needs to be authenticated to ensure the privacy of the users.
FR-5	Legal Requirements	Proper Medical Certificate is produced to ensure the integrity of the users.

❖ Non-functional Requirements:

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

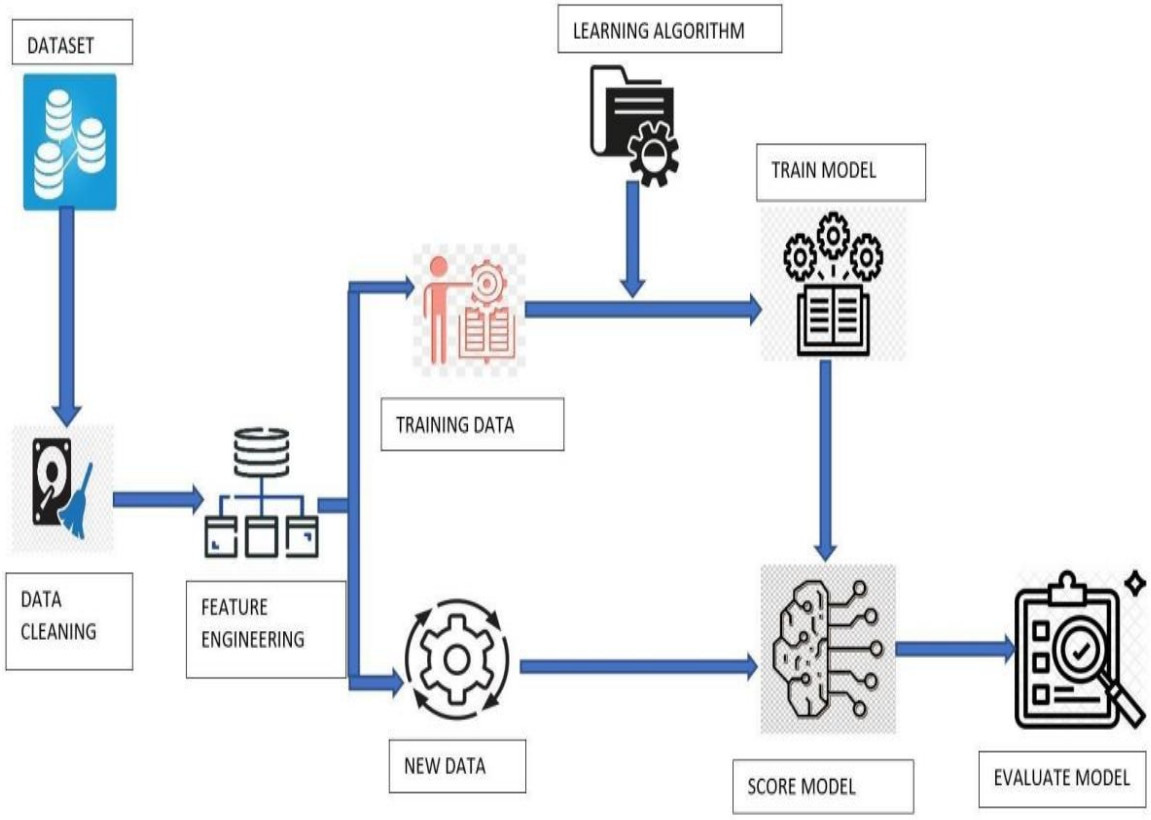
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The designed system is easy to use for specially abled persons as it is portable and platform independent.
NFR-2	**Security**	The system should protect the users data in a secure manner and avoid eavesdropping and such activities by means of encryption and decryption. The users will have a password which helps in secure login.
NFR-3	**Reliability**	The system is tested with large number of data in order to maintain the reliability of the users which is needed most on now a day.
NFR-4	**Performance**	The response time should be faster that improves the performance which is essential to cope up with the challenging world.
NFR-5	**Availability**	The system is available on 24/7 to use only the internet is needed for effective communication.
NFR-6	**Scalability**	The designed system should need to increase its performance whenever need is more and response to changes in processing demands.

PROJECT DESIGN

Data Flow Diagrams:

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

Flow Diagram



Technology Stack

Technical Architecture:

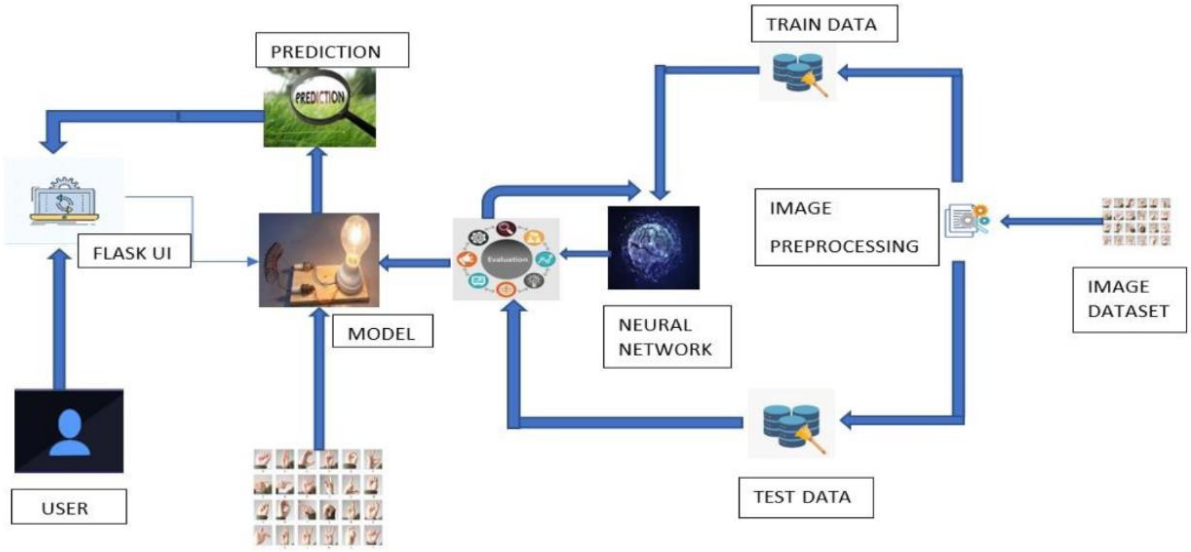


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	It deals with variety of frameworks, libraries and supports required to develop the project	Java / Python
3.	Application Logic-2	Helps in converting human voice into written words, In simple it is used to convert speech to text.	IBM Watson STT service
4.	Application Logic-3	Provides fast ,consistent and accurate answers during the execution phase of the project	IBM Watson Assistant
5.	Database	It can be numerical, categorical or time-series data	MySQL, NoSQL, etc.
6.	Cloud Database	Enables the user to use host database without buying the additional hardware	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage should be highly flexible, scalable and effective	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Used to access the information in the cloud	IBM Weather API, etc.
9.	External API-2	Used to access the information for data driven decision making	Aadhar API, etc.
10.	Machine Learning Model	Machine Learning Model deals with various algorithms that are needed for the implementation	Real time communication using AI for specially abled
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Install the windows version and execute the installer Select APACHE to install web server	Local, Cloud Foundry, Kubernetes, etc.

		Cloud Server Configuration : This server deals with the additional storage	

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The frameworks used are	Tensor flow, Theano, RNN, PyTorch, Caffle 2
2.	Security Implementations	the security / access controls implemented, use of firewalls etc.	Identify, Prevent aznd Respond
3.	Scalable Architecture	the scalability of architecture (3 – tier, Micro-services)	Data , models, operate at size, speed and complexity
4.	Availability	the availability of application (e.g. use of load balancers, distributed servers etc.)	Image and facial recognition, lip reading, text summarization, real time captioning
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Full and effective participation , equality of opportunity, accessibility

USER STORIES

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard.	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm.	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook.	I can register & access the dashboard with Facebook Login.	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail.	I can register the application using gmail with details linked to the gmail.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password.	Can enter these credentials either by manual or by auto filling depends on the case.	High	Sprint-1
	Dashboard	USN-6	As a user, I want to know about my data which I have given to see them visually appealing.	Can see the user data after successfully logging onto the application.	Medium	Sprint-1
Customer (Web user)	Registration	USN-7	As a User, I can register for the application through web by entering mobile number / gmail, password and confirming it.	I can access my account / dashboard through web.	High	Sprint-1
Customer Care Executive		USN-8	As a user , I Can get any support if needed by dialing the call or clicking the support.	After completing the registration the user can avail this service.	Medium	Sprint-1
Administrator		USN-9	The company should take care of the admin functionalities.	Admin should have access to each information registered by the user.	High	Sprint-1
Sign Up		USN-10	As a User, I should need to sign up if I don't haven't registered for the account earlier.	The credentials used for signing up should be unique.It is not used by	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
				any other users.		
Wishlist		USN-11	As a user, I am before avail for the service which can be kept aside.	As a User, I can review and use the services if needed.	Low	Sprint-2
Enrolled		USN-12	As a user, I can use the service after enrolling.So that the user can know and use.	As a user, it is quite appealing.	Low	Sprint-2

PROJECT PLANNING AND SCHEDULING

Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirements	User story number	User Story/Task	Story Points	Priority	Team Members
Sprint-1		USN-1				
Sprint-1		USN-2				
Sprint-2		USN-3				
Sprint-2		USN-4				
Sprint-3	Front-end	USN-5	Create front end for all above listed services and integrate them with the model.	5	Medium	Shiva Raghavesh
Sprint-3	Front-end	USN-6	Create front end for all above listed services and integrate them with the model.	5	Medium	Shiva Raghavesh
Sprint-4	Deploy the appliction	USN-7		5	Medium	Shiva Raghavesh
Sprint-4	Additional features	USN-8	Implement all additional features of the application	5	Low	Padmesh Partheban
Sprint-4	Testing	USN-9	Testing all features of the application	15	High	Padmesh Partheban

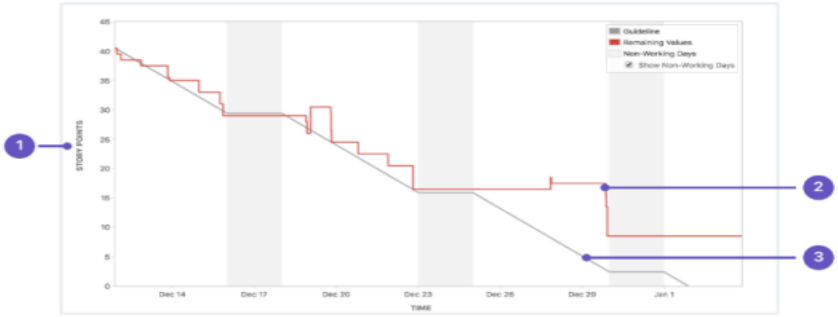
Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	6 Days	23 Oct 2022	29 Oct 2022	Will be updated as we go.	
Sprint-2	13	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	25	6 Days	14 Nov 2022	19 Nov 2022		

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = (15 + 13 + 10 + 25)/6 = 10.5$$

Burndown Chart:



1 **Estimation statistic**: The vertical axis represents the estimation statistic that you've selected.

2 **Remaining values**: The red line represents the total amount of work left in the sprint, according to your team's estimates.

3 **Guideline**: The grey line shows an approximation of where your team should be, assuming linear progress. If the red line is below this line, congratulations - your team's on track to completing all their work by the end of the sprint. This isn't foolproof though; it's just another piece of information to use while monitoring team progress.

CODING AND SOLUTIONING

Feature 1:

A real time image given as input is preprocessed converting it into a form similar to the data images used for training the model. This feature focusses on predicting the label of the image given as input.

Code:

```
import os
import json
from flask import Flask, redirect, url_for, render_template, request, flash
from werkzeug.utils import secure_filename
app = Flask(_name_,static_folder="static")
home_dir = os.getcwd()
UPLOAD_FOLDER = os.path.join(home_dir, "static")
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
data = []

import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import cv2
import pixellib
from pixellib.tune_bg import alter_bg
ans=""
change_bg = alter_bg()
change_bg.load_pascalvoc_model("deeplabv3_xception_tf_dim_orderin
g_tf_kernels.h5")
model = load_model('asl_model_97_56.h5')


@app.route('/',methods = ['GET'])
```

```
def hello_world():
    print("df")
    return render_template('index.html')

@app.route('/',methods = ['POST'])
def result():
    global ans
    if request.form.get('predict') == 'Predict’:
        return render_template('index.html',msg =
request.files['pic'].filename,images = json.dumps(data),msg1 = ans)
    else:
        if 'pic' not in request.files:
            print(ans)
        else:
            file = request.files['pic']
            file_name = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'],
file_name))
            data.append(file_name)
            change_bg.color_bg('static/'+file.filename, colors =
(255,255,255), output_image_name="colored_bg.jpg")

            img = cv2.imread('colored_bg.jpg',2)
            img=cv2.resize(img,(128,128))
            ret, bw_img =
cv2.threshold(img,254,255,cv2.THRESH_BINARY_INV)
            cv2.imwrite("masked.jpeg",bw_img)

img=image.load_img(r'masked.jpeg',target_size=(128,128),color_mode=
'grayscale')
            x=image.img_to_array(img)
            x=np.expand_dims(x,axis=0)
            pred=np.argmax(model.predict(x))
            temp=np.expand_dims(bw_img,axis=0)
            bw_img.shape
            index=['A','B','C','D','E','F','G','H','I']
            ans += index[pred] + " "
```

```
    return render_template('index.html',msg =
request.files['pic'].filename,images = json.dumps(data))

if _name_ == '_main_':
   app.run()
```

Feature 2:

In this feature, the predicted letter or a combination of letters are given as output in the form of audio. Text passed to the ‘parallel’ function below,is given as audio output.

Code:

```
import concurrent.futures
import sys
import pyttsx3
from time import sleep


def typing(text):
    for char in text:
        sleep(0.04)
        sys.stdout.write(char)
        sys.stdout.flush()

def textToSpeech(text):
    engine = pyttsx3.init()
    voices = engine.getProperty('voices')
    engine.setProperty('voice', voices[0].id)
    engine.setProperty('rate', 250)
    engine.say(text)
    engine.runAndWait()
    del engine
```

```
def parallel(text):
    with
concurrent.futures.ThreadPoolExecutor(max_workers=2) as
executor:
        future_tasks = {executor.submit(textToSpeech, text),
executor.submit(typing, text)}
        for future in
concurrent.futures.as_completed(future_tasks):
            try:
                data = future.result()
            except Exception as e:
                print(e)
```

TESTING

1.Purpose of document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3.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	51	0	0	51
Security	2	0	0	2
Exception reporting	9	0	0	9
Final report	4	0	0	4
Version control	2	0	0	2

PERORMANCE METRICS

Performance Metrics

NFT - Risk Assessment

S.No	Project Name	Scope/Feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume changes	Risk Score	Justification
1	Image Upload	New	High	No Changes	High	Nil	>10 to 30%	ORANGE	Image Upload is crucial to the platform
2	Text Result	New	High	No Changes	High	Nil	>10 to 30%	RED	Text Result is the core of the platform
3	Voice Result	New	High	No Changes	High	Nil	>10 to 30%	GREEN	Voice result is optional requirement

NFT - Detailed Test Plan

S.No	Project Name	NFT Test approach	Assumption/Dependencies/Risks	Approals/Signoff
1	Real time communication for physically disabled	End to end manual testing	1. The user knows to operate a browser. 2. The user has internet. 3. The user only selects the domains they like	Approved

ADVANTAGES

1) Given a real time image , it is converted to binary image similar to the data set used for training , hence giving higher accuracy
2) Alphabets are detected accurately , first stepping stone in the communication between specially abled people
3) The user interface is simple , one need to just upload the sign he/she needs to communicate
4) The predicted output is highly accurate and is given both in text and audio format.

DISADVANTAGES

1) Each letter need a sign , so it becomes time consuming for the people using it for communication.
2) This requires realtime image capturing under plain background(gives higher accuracy) , which is not the case most of the times.
3) This project is built to detect only alphabet , implementing the same to detect word / sentences will be more useful.

CONCLUSION

This project is built to help the specially abled people , who feel isolated from the society being unable to communicate. This is the first stepping stone in the communication of the specially abled people. The alphabets detected here can be enhanced to predict words which makes the communication much easier ,faster and more natural .

FUTURE SCOPE

1) The model done now predicts letters acuurately , this can be extended to predicting words by changing label and dataset fed to the model.
2) The model can be fine tuned to work for any background(not just plain) by using advanced image processing techniques. This makes it more realistic and use from anywhere.
3) Mobile application can be built for the same making it accessible to all and convenient.
4) The audio output can be trained to give output in sentences making it understandable.
5) UI can be extended to make it more appealing and user friendly

APPENDIX

Source Code

training.py

Loading the Dataset & Image Data Generation

```
[1]
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```
Python

```
[2]
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```
Python

```
[4]
# Training Dataset
x_train=train_datagen.flow_from_directory(r'C:\IBM\Dataset\training_set',target_size=(128,128), class_mode='categorical',batch_size=200,color_mode='grayscale')
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'C:\IBM\Dataset\test_set',target_size=(128,128), class_mode='categorical',batch_size=200,color_mode='grayscale')
```
Python

```
Found 15750 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

```
[5]
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```
Python

```
Len x-train :  79
Len x-test :  12
```

```
[6]
# The Class Indices in Training Dataset
x_train.class_indices
```
Python

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

Model Creation

```
[7]
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```
Python

```
[8]
# Creating Model
model=Sequential()
```
Python

```
[9]
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(128,128,1)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

# Adding Hidden Layers
model.add(Dense(100,activation='relu'))
model.add(Dense(100,activation='relu'))
# Adding Output Layer
model.add(Dense(9,activation='softmax'))
```
Python

```
[10]
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```
Python

```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=3,validation_data=x_test,validation_steps=len(x_test))
```

Python

```
C:\Users\nchel\AppData\Local\Temp\ipykernel_10068\1200875117.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`,
which supports generators.
  model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=3,validation_data=x_test,validation_steps=len(x_test))

Epoch 1/3
79/79 [==============================] - 50s 636ms/step - loss: 0.0222 - accuracy: 0.9938 - val_loss: 0.2424 - val_accuracy: 0.9662
Epoch 2/3
79/79 [==============================] - 50s 628ms/step - loss: 0.0162 - accuracy: 0.9956 - val_loss: 0.1718 - val_accuracy: 0.9724
Epoch 3/3
79/79 [==============================] - 50s 630ms/step - loss: 0.0136 - accuracy: 0.9973 - val_loss: 0.1376 - val_accuracy: 0.9756

<keras.callbacks.History at 0x12e321d1e50>
```

Saving the Model

```
model.save('asl_model_97_56.h5')
```

Python

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

Python

```
img=image.load_img(r'C:\IBM\Dataset\test_set\E\2.png',target_size=(128,128),color_mode='grayscale')
```

Python

```
img
```

Python

```
x=image.img_to_array(img)
x.ndim
x=np.expand_dims(x,axis=0)
x.ndim
```

Python

```
4
```

```
pred=np.argmax(model.predict(x))
```

Python

```
1/1 [==============================] - 0s 62ms/step
```

```
index=['A','B','C','D','E','F','G','H','I']
print(index[pred])
```

Python

```
E
```

hello.py

```
import os
import json
from flask import Flask, redirect, url_for, render_template, request, flash
from werkzeug.utils import secure_filename
app = Flask(__name__,static_folder="static")
home_dir = os.getcwd()
UPLOAD_FOLDER = os.path.join(home_dir, "static")
```

```
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
data = []

import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import cv2
import pixellib
from pixellib.tune_bg import alter_bg
ans=""
change_bg = alter_bg()
change_bg.load_pascalvoc_model("deeplabv3_xception_tf_dim_ordering_tf_kernels.h5"
)
model = load_model('asl_model_97_56.h5')

import concurrent.futures
import sys
import pyttsx3
from time import sleep


def typing(text):
    for char in text:
        sleep(0.04)
        sys.stdout.write(char)
        sys.stdout.flush()

def textToSpeech(text):
    engine = pyttsx3.init()
    voices = engine.getProperty('voices')
    engine.setProperty('voice', voices[0].id)
    engine.setProperty('rate', 250)
    engine.say(text)
    engine.runAndWait()
    del engine

def parallel(text):
    with concurrent.futures.ThreadPoolExecutor(max_workers=2) as executor:
        future_tasks = {executor.submit(textToSpeech, text),
executor.submit(typing, text)}
        for future in concurrent.futures.as_completed(future_tasks):
            try:
                data = future.result()
            except Exception as e:
                print(e)
```

```
# parallel("Speak this!")
# sleep(4.0)

@app.route('/',methods = ['GET'])
def hello_world():
    print("df")
    return render_template('index.html')

@app.route('/',methods = ['POST'])
def result():
    global ans
    if request.form.get('predict') == 'Predict':
        parallel(ans)
        return render_template('index.html',msg =
request.files['pic'].filename,images = json.dumps(data),msg1 = ans)
    else:
        if 'pic' not in request.files:
            print(ans)
        else:
            file = request.files['pic']
            file_name = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], file_name))
            data.append(file_name)
            change_bg.color_bg('static/'+file.filename, colors = (255,255,255),
output_image_name="colored_bg.jpg")

            img = cv2.imread('colored_bg.jpg',2)
            img=cv2.resize(img,(128,128))
            ret, bw_img = cv2.threshold(img,254,255,cv2.THRESH_BINARY_INV)
            cv2.imwrite("masked.jpeg",bw_img)
            img=image.load_img(r'masked.jpeg',target_size=(128,128),color_mode='g
rayscale')
            x=image.img_to_array(img)
            x=np.expand_dims(x,axis=0)
            pred=np.argmax(model.predict(x))
            temp=np.expand_dims(bw_img,axis=0)
            bw_img.shape
            index=['A','B','C','D','E','F','G','H','I']
            ans += index[pred] + " "
    return render_template('index.html',msg =
request.files['pic'].filename,images = json.dumps(data))

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

dummy.py

```
import cv2
import mediapipe as mp

mphands = mp.solutions.hands
hands = mphands.Hands()
mp_drawing = mp.solutions.drawing_utils
cap = cv2.VideoCapture(0)
# while True:
#     if(cap.isOpened()):

#         break

_, frame = cap.read()
# h,w=500,500
h, w, c = frame.shape
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import cv2
import pixellib
from pixellib.tune_bg import alter_bg

change_bg = alter_bg()
change_bg.load_pascalvoc_model("deeplabv3_xception_tf_dim_ordering_tf_kernels.h5"
)
model = load_model('asl_model_97_56.h5')

while True:
    # while(cap.isOpened()):
        _, frame = cap.read()
        # frame = cv2.flip(frame, 1   )
        framergb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        # frame
        result = hands.process(frame)
        hand_landmarks = result.multi_hand_landmarks
        if hand_landmarks:
            for handLMs in hand_landmarks:
                x_max = 0
                y_max = 0
                x_min = w
                y_min = h
```

```
                for lm in handLMs.landmark:
                    x, y = int(lm.x * w), int(lm.y * h)
                    if x > x_max:
                        x_max = x
                    if x < x_min:
                        x_min = x
                    if y > y_max:
                        y_max = y
                    if y < y_min:
                        y_min = y
                if(x_max-x_min>y_max-y_min):
                    y_max+=(x_max-x_min-y_max+y_min)//2
                    y_min-=(x_max-x_min-y_max+y_min)//2
                else:
                    x_max+=(y_max-y_min-x_max+x_min)//2
                    x_min-=(y_max-y_min-x_max+x_min)//2
                cv2.rectangle(framergb, (x_min-30, y_min-30), (x_max+30,
y_max+30), (0, 255, 0), 2)
                cv2.imwrite("testbox.jpg",framergb[max(0,y_min-
30):min(y_max+30,h),max(x_min-30,0):min(x_max+30,w)])
                change_bg.color_bg("testbox.jpg", colors = (255,255,255),
output_image_name="colored_bg.jpg")

                img = cv2.imread('colored_bg.jpg',2)
                img=cv2.resize(img,(128,128))
                ret, bw_img = cv2.threshold(img,254,255,cv2.THRESH_BINARY_INV)
                cv2.imwrite("masked.jpeg",bw_img)
                img=image.load_img(r'masked.jpeg',target_size=(128,128),color_mod
e='grayscale')
                x=image.img_to_array(img)
                x.ndim
                x=np.expand_dims(x,axis=0)
                pred=np.argmax(model.predict(x))
                temp=np.expand_dims(bw_img,axis=0)
                bw_img.shape
                index=['A','B','C','D','E','F','G','H','I']
                print(index[pred])
                # mp_drawing.draw_landmarks(frame, handLMs,
mphands.HAND_CONNECTIONS)
        cv2.imshow("Frame", framergb)

        cv2.waitKey(1)
```

Frontend

```
<html style="text-align:center">
    <h1 style = "text-align: center;">Real Time Communication using AI for
specially abled</h1>
    <link rel="stylesheet" href="/static/index.css">
    <div id = "gallery" style="margin:10px">
    </div>
    <script>
        function test_func(data) {
            for(i=0;i<data.length;i++){
                item = document.createElement("img");
                const x = `../static/${data[i]}`
                item.setAttribute("src",x);
                item.setAttribute("height","200");
                item.setAttribute("width","200");
                item.setAttribute("margin","10")
                item.setAttribute("border","2px dashed black")
                document.getElementById("gallery").appendChild(item);
            }
        }
        test_func({{ images|safe }})
    </script>
    <div>
        <form action = "http://localhost:5000/" method="post"
enctype="multipart/form-data" id="form" style="border:2px solid black;
        box-shadow: 5px 5px 5px;text-align:center;padding:10px;margin:0 auto;
width:600px;height:300px">
            <br/><br/>
            <input style="padding:10px;" type="file" name="pic" value="Upload
image">
            <br/><br/>
            <input style="padding:10px;" type="submit" name = "upload"
value="Upload">
            <br/><br/>
            <input style="padding:10px;" type="submit" name = "predict"
value="Predict">
            </form>
    </div>
    <div id = "ans">
        <h2 style="color:green;">{{ msg1 }}</h2>
    </div>
</html>
```

GITHUB LINK – https://github.com/IBM-EPBL/IBM-Project-17352-1659635284

DEMO LINK-
https://drive.google.com/file/d/1Aqyo_98sFSW5xZQCzIP65bcVAibY9VRN/view?usp=sharing