PROJECT REPORT

REAL–TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

TEAM ID : PNT2022TMID44066

Submitted by

GAYATHRI.G
PAVITHRA.S
ABINANDHINI.R
VISHNUPRIYA.S
SATHIYA PRIYA.B

ABSTRACT:

One of the important problems that our society faces is that people with disabilities are finding it hard to cope up with the fast growing technology. The access to communication technologies has become essential for the handicapped people. Generally deaf and dumb people use sign language for communication but they find difficulty in communicating with others who don't understand sign language. Sign language is an expressive and natural way for communication between normal and dumb people (information majorly conveyed through the hand gesture). So, we need a translator to understand what they speak and communicate with us. The sign language translation system translates the normal sign language to speech and hence makes the communication between normal person and dumb people easier. So, the whole idea is to build a communication system that enables communications between speech-hearing impaired and a normal person.

1.INTRODUCTION:

1.1 Overview:

Communication is important to express oneself. It also satisfies one's needs. One should have effective communication for advancement in the career. In your personal life, effective communication skills can smooth your way and your relationships with others by helping you to understand others, and to be understood. 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.

1.2 Purpose:

An app is built which uses this model. The main purpose of the app is to enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

2.LITERATURE SURVEY:

2.1 Existing problem:

To facilitate easier communication for specially abled people with normal people by developing a model that incorporates necessary features including sign language interpretation and classificationSome of the existing solutions for solving this problem are:

Technology:

One of the easiest ways to communicate is through technology such as a smart phone or laptop. A deaf person can type out what they want to say and a person who is blind or haslow vision can use a screen reader to read the text out loud. A blind person can also use voice recognition software to convert what they are saying in to text so that a person who is Deaf can then read it.

Interpreter:

If a sign language interpreter is available, this facilitates easy communication if the person who is deaf is fluent in sign language. The deaf person and person who is blind can communicate with each other via the interpreter. The deaf person can use sign language and the interpreter can speak what has been said to the person who is

blind and then translate anything spoken by the blind person into sign language for the deaf person.

Just Speaking:

Depending on the deaf person's level of hearing loss, they may be able to communicate with a blind person who is using speech. For example, a deaf person may have enough residual hearing (with or without the use of an assistive hearing device such as a hearing aid) to be able to decipher the speech of the person who is blind or has low vision. However, this is often not the most effective form of communication, as it is very dependent on the individual circumstances of both people and their environment (for example, some places may have too much background noise).

2.2 Reference:

- AUTHOR: K. Sunitha, Anitha Saraswathi, Aarthi, Jaya Priya, Lingam Sunny Year:2016. Publication: republication Title: Deaf Mute Communication Interpreter.

- AUTHOR: Anbarasi Rajamohan, Hemavathy R, Dhana Lakshmi M. Year:2013, Publication: International Journal of Scientific Engineering and Technology Title: Deaf-Mute Communication Interpreter.

- AUTHOR: Kedar Potdar, Gauri Nagavkar Year:2017, Publication: computer science Title: Real-time Communication System for the Deaf and Dumb.

- AUTHOR: Alex Rupom Hasdak, Istiaq Al Nur, Adnan Al Neon, Hasan U. Zaman. Year:2018 Publication: IEEE Title: Deaf-Vibe: A Vibrotactile Communication Device Based on Morse Code for Deaf-Mute Individuals.

- AUTHOR: B.Jadhav, Nipun Munot, Madhura Hambarde, Jueli Ashtikar Year:2015, Publication: republication Title: Hand Gesture Recognition to Speech Conversion in Regional Language.

2.3 Problem statement Definition:

This project aims to aid the deaf-mute by creating a new system that helps convert sign language to text and speech for easier communication with the audience. Communication 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 in hand sign language. In emergency times conveying their message is very difficult.

To facilitate easier communication for specially abled people with normal people by developing a model that incorporates necessary features including sign language interpretation and classification.

5 W's:

Who does the problem affect?

- Specially abled people and their family members.
- Doctors and other healthcare staffs treating the specially abled people.
- Caretakers present in the special care homes.
- A normal person when encountering a specially abled person.

What is the issue?

The lack of proper communication facilities to convey a wide variety of messages between a specially abled and a normal person who doesn't know sign language.

When does the issue occur?

- When there is a need to convey an urgent message.
- When encountering new people and places.

Where is the issue occurring?

The issue can happen wherever a specially abled person has an access but lacks services to assist them.

Why is it important that we fix the problem?

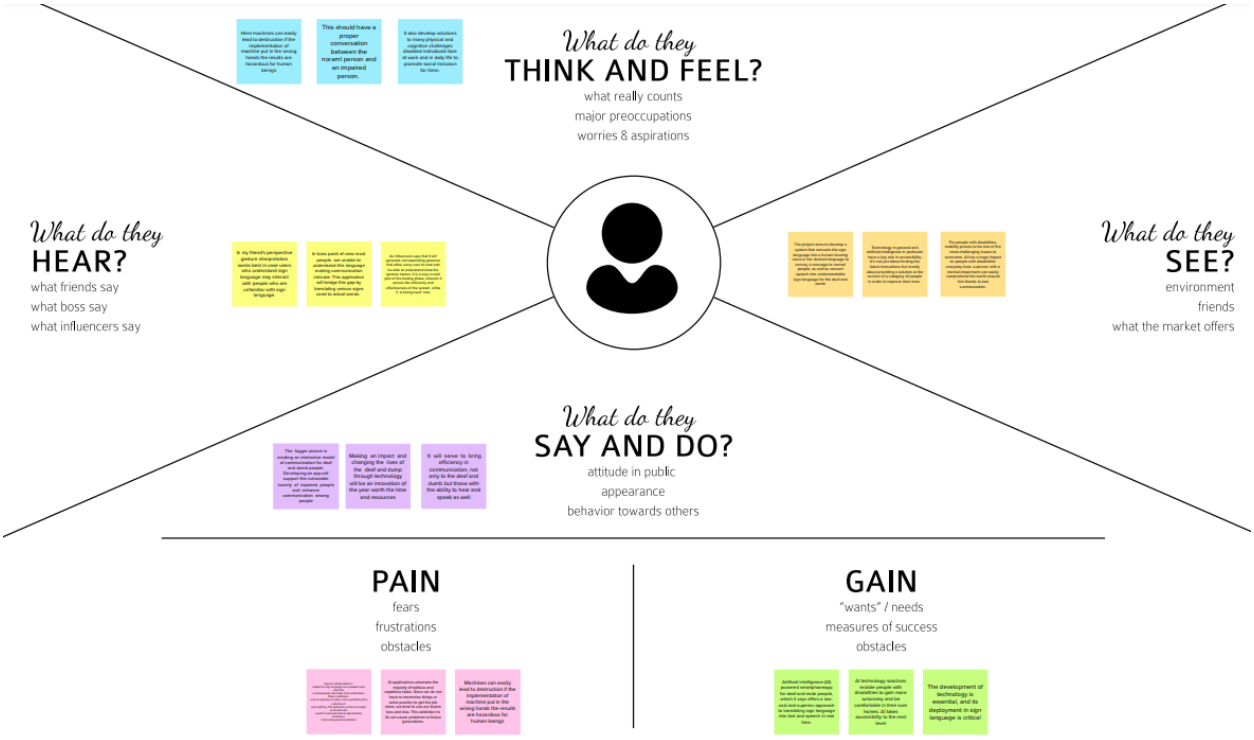
According to a number of study reports, it has been witnessed that the numbers of depressed specially abled persons are quite increasing due to the difficulties they

face during their communication with normal people. With the advent of current technological facilities, it is possible to stop the continuation of this trend

3. IDEATION AND PROPOSED SOLUTION:

3.1 Empathy Map Canvas:

An empathy map helps to map what a design team knows about the potential audience. This tool helps to understand the reason behind some actions a user takes deeply. This tool helps build Empathy towards users and helps design teams shift focus from the product to the users who are going to use the product. By using the empathy map we can able to understand the review of the audience.



3.2 Ideation and brainstorming:

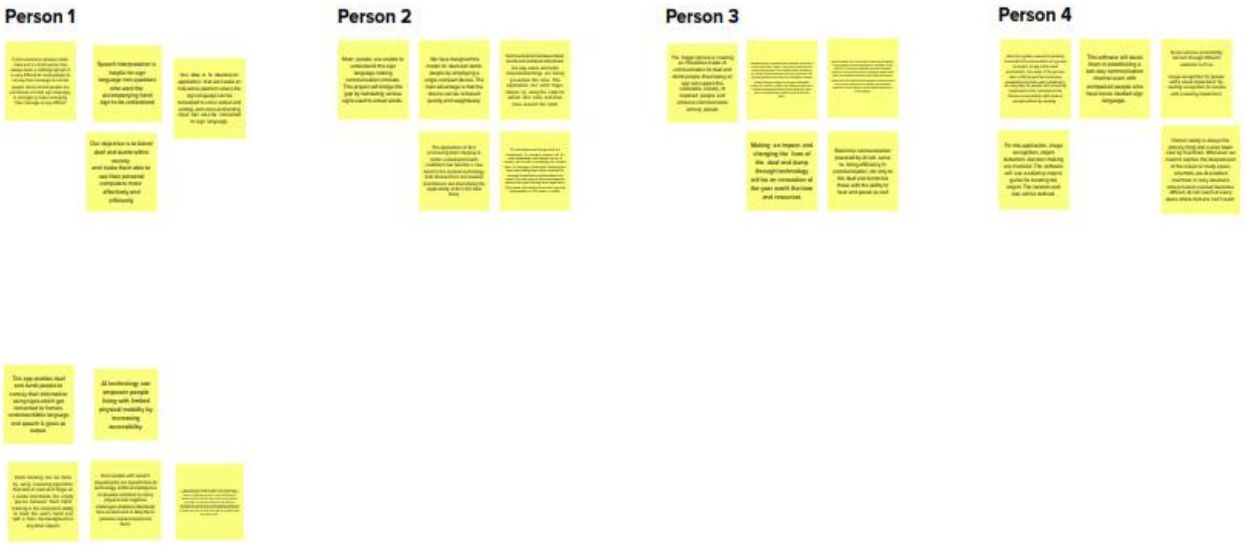
Brainstorming is a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind.

- AI technology solutions enable people with disabilities to gain more autonomy and be comfortable in their own homes. AI takes accessibility to the next level.

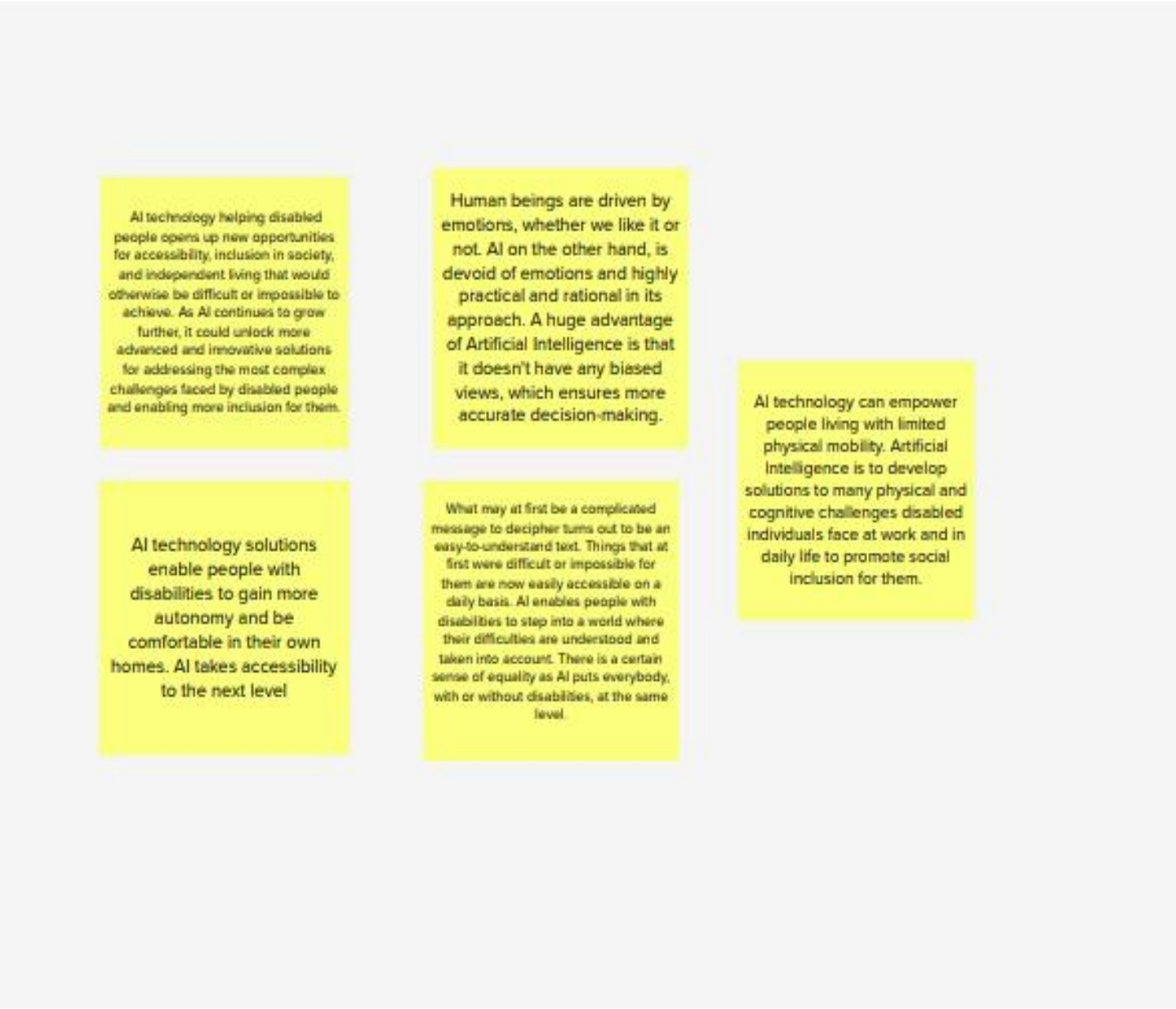
- This app enables deaf and dumb people to convey their information using signs which get converted to human understandable language and speech is given as output.

- The gestures are viewed on display. The image will be recognized through a web camera. If the system detects some unrecognizable gestures, it will automatically be refreshed for the user to be able to make proper gestures once again. This artificial intelligence that was being developed can identify errors on hand gesture matches and will stop as a default. It will generate corresponding gestures that allow every user to read and be able to understand what the gesture mean.

BRAINSTORMING:



GROUP IDEAS:


AI technology helping disabled people opens up new opportunities for accessibility, inclusion in society, and independent living that would otherwise be difficult or impossible to achieve. As AI continues to grow further, it could unlock more advanced and innovative solutions for addressing the most complex challenges faced by disabled people and enabling more inclusion for them.
Human beings are driven by emotions, whether we like it or not. AI on the other hand, is devoid of emotions and highly practical and rational in its approach. A huge advantage of Artificial Intelligence is that it doesn't have any biased views, which ensures more accurate decision-making.
AI technology can empower people living with limited physical mobility. Artificial Intelligence is to develop solutions to many physical and cognitive challenges disabled individuals face at work and in daily life to promote social inclusion for them.
AI technology solutions enable people with disabilities to gain more autonomy and be comfortable in their own homes. AI takes accessibility to the next level
What may at first be a complicated message to decipher turns out to be an easy-to-understand text. Things that at first were difficult or impossible for them are now easily accessible on a daily basis. AI enables people with disabilities to step into a world where their difficulties are understood and taken into account. There is a certain sense of equality as AI puts everybody, with or without disabilities, at the same level.

PRIORITIZE:

3.3 Proposed solution:

This paper describes the system that overcomes the problem faced by the speech and hearing impaired. The objectives of the research are as follows:

1)To design and develop a system which lowers the communication gap between speech-hearing impaired and normal world.

2)To build a communication system that enables communications between deaf-dumb person and a normal person.

3)The main approaches for analyzing and classifying hand gestures for HCI include Glove based techniques and Vision based techniques.

The proposed system consists of mainly four processes i.e. Image acquisition, Image preprocessing, Feature extraction and Image classification.

A. Image Acquisition

Image acquisition is the process to capture the hand gesture images which represent different signs. The resolution of various image capturing devices may not be the same. This results in different resolutions of the captured images. For accurate comparison of the features and to reduce the computational effort needed for processing. All the images should be scaled to a uniform size. Thus the images for training and testing are captured in a white background with a web camera and a database is created. This database consists of 26 hand gestures. The system works offline recognition i.e. we give a test image as input to the system and the system tells us which sign is recognized .

B. Preprocessing

A Preprocessing is a very much required task to be done in hand gesture recognition system. Preprocessing is applied to images before we can extract features from hand images. The main goal of image segmentation is domain independent partitioning of an image into a set of disjoint regions that are visually different, homogeneous and meaningful with respect to some characteristics or computed property such as grey level, texture or colour to enable easy image analysis. Image segmentation is a useful tool in many realms including industry, health care, astronomy, and various other fields.Segmentation in concept is a very simple idea. Simply looking at an image, one can tell what regions are contained in a picture. Is it a building, a person, a cell, or just simply background? Visually it is very easy to determine what a region of interest is and what is not. Doing so with a

computer algorithm on the other hand is not so easy. The image segmentation results are a set of regions that cover the entire image together and a set of contours extracted from the image. All of the pixels in a region are similar with respect to some characteristics such as color, intensity, or texture. Adjacent regions are considerably different with respect to the same individuality. Segmentation is the classification of the input colored image into skin and non-skin pixels based on skin color information. A wide range of applications that require the segmentation process as a preprocessing operation such as computer vision, face/ hand detection and recognition, medical image analysis, and pattern recognition. Color information is one of the simple cues used for detecting skin color, and the use of proper color space to represent color information of an image is a crucial decision. The selection of appropriate segmentation method depends on the application and system environments. The performance of any segmentation algorithm is quantified using some benchmarking such as recall and precision coefficients, or by calculating the percentage of correct and false detection rates according to the complexion of the technique used.

C. Feature Extraction

Good segmentation process leads to perfect features extraction process and the latter play an important role in a successful recognition process. There are many interesting points on every object which can be extracted to provide a "feature" description of the object. Features vectors of the segmented image can be extracted in different ways according to particular application. Under different scene conditions, the performance of different feature detectors will be significantly different. The nature of the background, existence of other objects (occlusion), and illumination must be considered to determine what kind of features can be efficiently and reliably detected . Feature extraction is a method of reducing data dimensionality by encoding related information in a compressed representation and removing less discriminative data. Feature extraction is vital to gesture recognition performance. The selection of which features to deal with and the extraction method are probably the most significant design decisions in hand motion and gesture recognition development.

D. Classification

Classification between the objects is an easy task for humans but it has proved to be a complex problem for machines. The rise of high-capacity computers, the availability of high quality and low priced video cameras, and the increasing need

for automatic video analysis has generated an interest in object classification algorithms. A simple classification system consists of a camera fixed high above the interested zone, where images are captured and consequently processed. Classification includes image sensors, image preprocessing, object detection, object segmentation, feature extraction and object classification. Classification system consists of a database that contains predefined patterns that compares with detected objects to classify into proper categories. Image classification is an important and challenging task in various application domains, including biomedical imaging, biometry, video surveillance, vehicle navigation, industrial visual inspection, robot navigation, and remote sensing . Image classification is perhaps the most important part of digital image analysis. Classification of remotely sensed data is used to assign corresponding levels with respect to groups with homogeneous characteristics, with the aim of discriminating multiple objects from each other within the image. The level is called class. Classification will be executed on the base of spectral or spectrally defined features, such as density, texture etc. in the feature space.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A Real time communication system needs some modifications before transferring different speech signals simultaneously on the same band/channel. The signals are sampled from analog audio signals with an introduction of quantization noise and sampling degradation.
2.	Idea / Solution description	More in recent times urbanized embedded applications are altering our standard of living in an elegant way. Sign language is a communicative and normal way for communication among common and the human being with the specially abled. The objective of the sign language translation system is to decode the usual sign language into speech and to create trouble-free contact with the dumb persons. The physical gesticulation statement comprises of hand gestures that suggest particular meaning, the non - physical is head movement, facial appearance, body orientation and position. The most important uniqueness of a person hand gesticulation can be reviewed by their energetic features. Hand gesture conversion to voice involves images of live hand gestures taken lively with which training datasets are developed and with this dataset the gestures are recognized. In the opposite process, the normal person voice has to be converted into gesture for the understanding of disabled persons. Speech recognition involves voice recognition and converting it into text and gesture images.

3.	Novelty / Uniqueness	Optimizing and improving the virtual communication for the specially abled people.

4.	Social Impact / Customer Satisfaction	Improving innovations and communication skill. Comfortable and Useful to communicate during at emergency situations.
5.	Business Model (Revenue Model)	Lack access to employment opportunities and even if they are able to get employment, they face problems such as reasonable accommodation at work, accessible public transportation to get them to work and back and discrimination and ignorance about their potential at work.
6.	Scalability of the Solution	The persons with disabilities are more likely to experience adverse socio-economic outcomes such as less education poorer health outcomes, lower levels of employment, and higher poverty rates. Poverty may increase the risk of disability through malnutrition, inadequate access to education and health care.

3.4 Proposed solution kit:

4. REQUIREMENTS ANALYSIS:

4.1 Functional requirements:

- Operating system – Windows, Macs, Linux
- CPU – Multi core Processor (i3 or above)
- CPU – NVIDIA AI capable of Google ‘s TPU
- Webcam – Integrated or external with full HD support

4.2 Non-functional requirements:

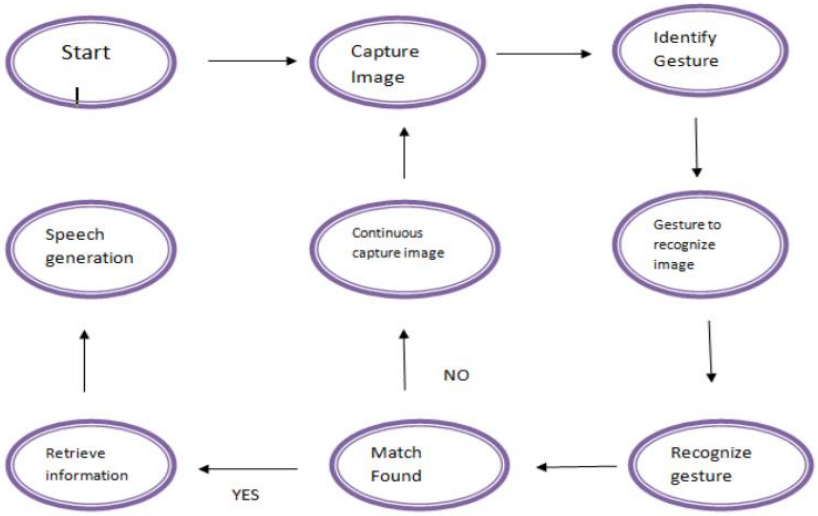
- Python – v3.9.0 or above
- Python packages – flask, tensor flow, open cv python, keras, NumPy, pandas, virtualenv, pillow
- Web browser – Mozilla Firefox, Google chrome or any modern web browser
- IBM cloud – Watson studio (Model training and deployment as Machine Learning Instance)

Si. No.	Components	Description	Technology
1.	User Interface	The user interface is the point of human computer interaction and communication in device.	Python flask, HTML, CSS/JavaScript.
2.	Flash UI	Flash's user interface components let you interact with the	Using the cloud, it can be executed.
		users that use your site and gather information.	
3.	Models	Support Vector Machine (SVM) is subsequently applied to classify our gesture image dataset.	Machine Learning.
4.	Image	Image processing is used to extract signs from the image using neural network.	ANN, CNN, Open CV.
5.	Evaluate data	Aims to estimate the generalization accuracy of a model on future (unseen/out-of-sample) data.	NLP.
6.	Unstructured data	Unstructured data is a conglomeration of many varied types of data that are stored in their native formats.	Natural Language Processing (NLP).
7.	Structured data	Typically categorized as quantitative data is highly organized and easily decipherable by machine learning algorithms.	Machine language and artificial intelligence tools.
8.	File Storage	File storage requirements to store the trained model in order to use it whenever it is needed.	IBM Block Storage or Cloud object.
9.	ML service	Provides a full range of tools and services so that you can build, train, and deploy Machine Learning models.	Python, IBM Watson.
10.	IBM Cloud	IBM Watson Studio empowers data scientists, developers and analysts to build, run and manage AI models, and optimize decisions anywhere on IBM Cloud Pak for Data	IBM Cloud and Watson Studio service
11.	Dataset	First prototype of this system used a dataset of 24 static signs from the Panamanian Manual Alphabet.	AI technology.

5. PROJECT DESIGN:

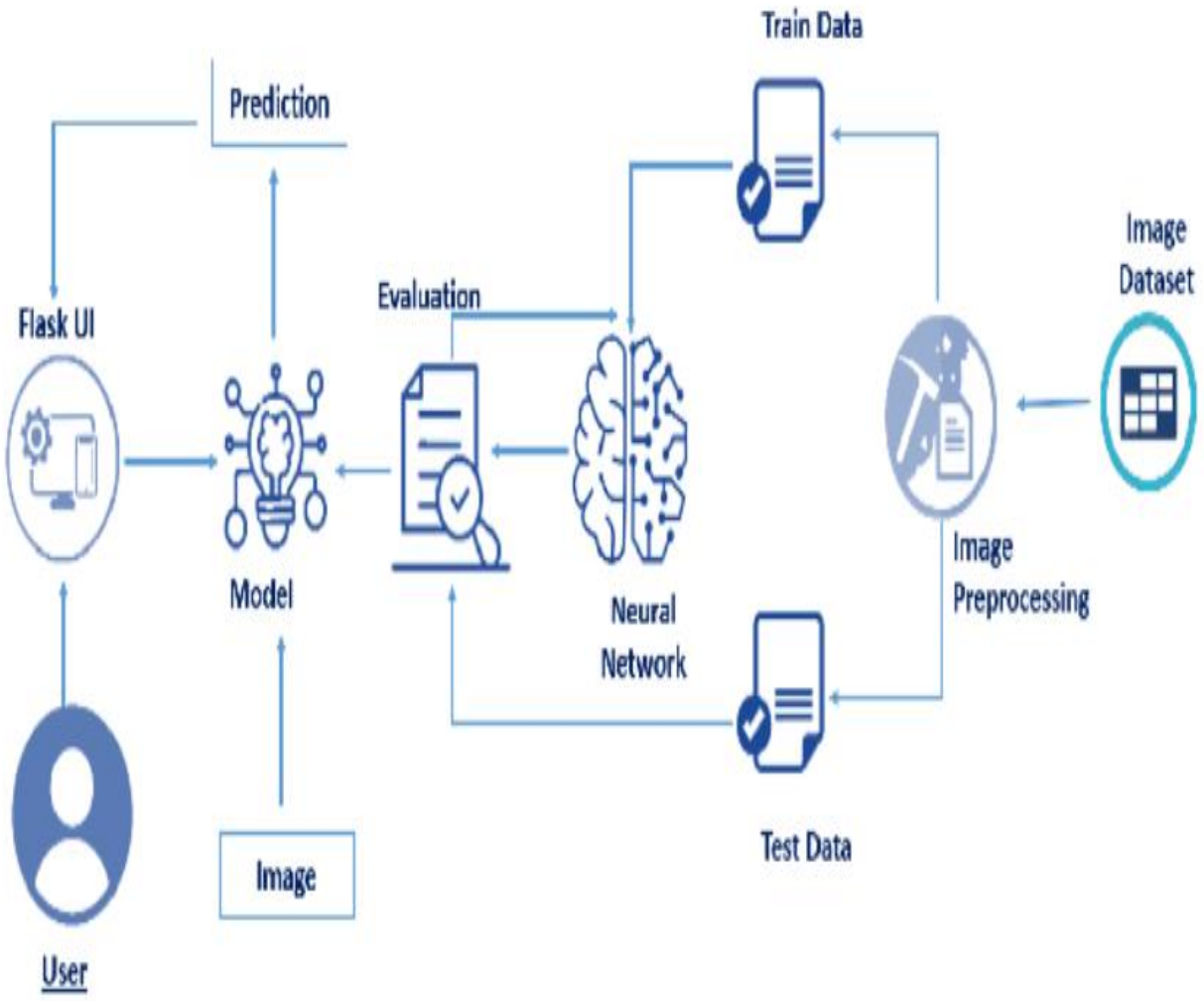
5.1 Data flow diagram:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored. A data flow diagram (DFD) maps out the flow of information for any process or system. 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. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.



DFDs help you better understand process or system operation to discover potential problems, improve efficiency, and develop better processes. They range from simple overviews to complex, granular displays of a process or system.

5.2 Solution and Technical Architecture:



5.3 User Stories:

User Stories

.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	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/criteria	High	Sprint-1
		USN-2	Once a user can register the application he/she receives the confirmation email.	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-2
	Dashboard	USN-4	Hearing impaired person perform sign that represent digit,alphabet,word or a sentence	User can perform the gesture sign in front of the camera for converting into speech recognition	High	Sprint-1
	Interface response	USN-5	As a user,I can say that gesture interpretation works best for the peoples who understand sign language may interact with people who are unfamiliar with sign language		High	Sprint-1
	Normal hearing person	USN-6	Normal hearing person is the passive user of the System		Medium	Sprint -2

6. PROJECT PLANNING AND SCHEDULING:

6.1 Sprint Planning and Estimation:

Milestones	Activities	Description
Ideation Phase	Literature Survey	Literature survey on the selected project & information gathering
	Empathy Map	Prepare Empathy map to capture the user Panis & Gains, prepare list of problem statement
	Ideation	Organizing the brainstroming session and prioritise the top 3 ideas based on feasibility & Importance
Project Design Phase I	Proposed Solution	Prepare proposed solution document which includes novelty, feasibility of ideas, business model, social impact, Scalability of solution
	Problem Solution Fit	Prepare problem solution fit document
	Solution Architecture	Prepare solution architecture document
Project Design Phase II	Customer Journey	Prepare customer journey map to understand the user interactions & experience with the application
	Functional requirement	Prepare functional & non functional requirement document
	Data Flow Diagram	Prepare Data Flow Diagram and user stories
	Technology architecture	Draw the technology architecture diagram
Project Planning Phase	Milestones & Activity list	Prepare milestones and activity list of the project
	Sprint Delivery Plan	Prepare sprint delivery plan

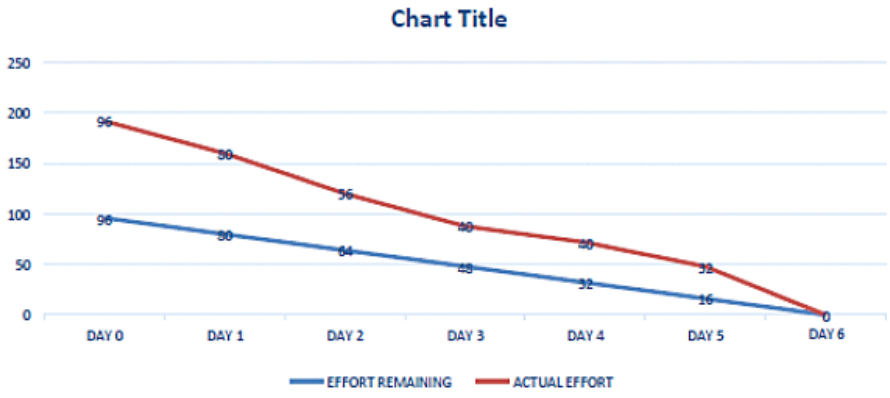
Milestones	Activities	Description
Project Development Phase	Delivery of Sprint – 1,2,3,4	To develop the code and submit the developed code by testing it
Setting up App environment	Create IBM Cloud account	Signup for an IBM Cloud account
	Create flask project	Getting started with Flask to create project
	Install IBM Cloud CLI	Install IBM Command Line Interface
	Docker CLI Installation	Installing Docker CLI on laptop
	Create an account in sendgrid	Create an account in sendgrid. Use the service as email integration to our application for sending emails
Implementing web Application	Create UI to interact with Application	Create UI • Registration page • Login page • URL text box • Displaying output
	Create IBM DB2 & connect with python	Create IBM DB2 service in IBM Cloud and connect with python code with DB
Deployment of App in IBM Cloud	Check the URL	The URL which needs to check should be entered on the column provided
	Compare using ML	The Processed dataset checks the given URL is phishing or not
	Output	By Comparing the given URL with the already defined one, the machine will produce the output as phishing website or not

6.2 Sprint Delivery Schedule:

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint – 1	Registration	USN – 1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High	GAYATHRI G ABINANDHINI R
Sprint – 1	Authentication	USN – 2	As a user, I will receive OTP to confirm details.	2	High	GAYATHRI G PAVITHRA S
Sprint – 1	Registration	USN – 3	As a user, I will receive confirmation email once I have registered for the application.	1	Low	PAVITHRA S
Sprint – 1	Login	USN – 4	As a user, I can log into the application by entering email & password.	2	High	PAVITHRA S VISHNUPRIYA S
Sprint – 2	Dashboard	USN – 5	As a user, I must have one place to explore all available features.	3	High	ABINANDHINI R SATHIYAPRIYA B
Sprint – 2	Login	USN – 6	As a user, If I forget my password, I must get an auto-generated password to reset my password.	2	Medium	VISHNU PRIYA S
Sprint – 3	Help	USN – 7	As a user, I must be able to reach out to the Support Team to get my issues resolved.	1	Low	GAYATHRI G
Sprint – 3	Management	USN – 8	As a user, I can access the site using mobile/ desktop.	3	High	PAVITHRA S SATHIYAPRIYA B
Sprint – 4	System	USN – 9	As a user, I must have access to previous usage history.	2	Medium	ABINANDHINI R
Sprint – 4	System	USN – 10	As a user, I can have audio output as well as text output.	3	High	VISHNU PRIYA S SATHIYAPRIYA B

6.3 Report from JIRA:

Burndown chart:



7. CODING AND SOLUTIONING:

437 lines (377 sloc) | 10.3 KB

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 40%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}
```

```css
span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
```

```
<script src="script.js"></script>
</body>
<html><head>
</head><body>
    <video src="" ></video>
    <br />
<button id='flipCamera'>Flip</button>
</body>
<script>
  var front = false;
var video = document.querySelector('video');
  document.getElementById('flipCamera').onclick = function() { front = !front;
  var constraints = { video: { facingMode: (front? "user" : "environment"), wi
  navigator.mediaDevices.getUserMedia(constraints)
  .then(function(mediaStream) {
    video.srcObject = mediaStream;
    video.onloadedmetadata = function(e) {
    video.play();
};
})
.catch(function(err) { console.log(err.name + ": " + err.message); })
</script></html>
</html>
<style>
.screenshot-image {
    width: 150px;
    height: 90px;
    border-radius: 4px;
    border: 2px solid whitesmoke;
    box-shadow: 0 1px 2px 0 rgba(0, 0, 0, 0.1);
    position: absolute;
    bottom: 5px;
    left: 10px;
    background: white;
}

.display-cover {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 70%;
    margin: 5% auto;
    position: relative;
}

video {
    width: 100%;
    background: rgba(0, 0, 0, 0.2);
}

.video-options {
    position: absolute;
    left: 20px;
    top: 30px;
}

.controls {
    position: absolute;
    right: 20px;
    top: 20px;
    display: flex;
}

.controls > button {
    width: 45px;
    height: 45px;
    text-align: center;
    border-radius: 100%;
```

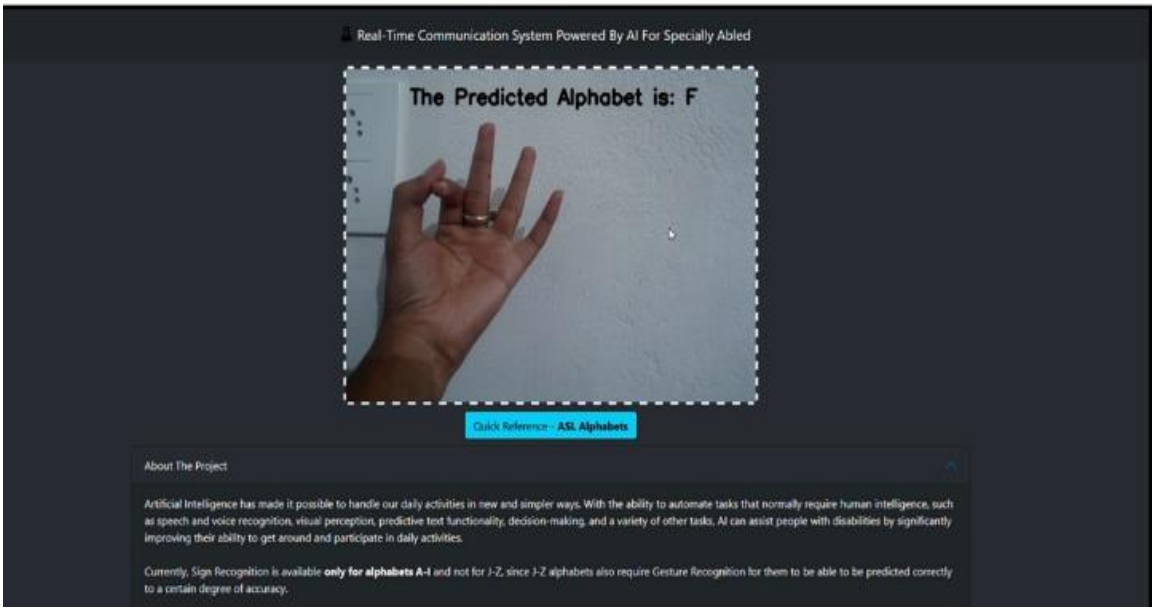
```
    margin: 0 6px;
    background: transparent;
}

.controls > button:hover svg {
    color: white !important;
}

@media (min-width: 300px) and (max-width: 400px) {
    .controls {
        flex-direction: column;
    }

    .controls button {
        margin: 5px 0 !important;
    }
}

.controls > button > svg {
    height: 20px;
    width: 18px;
    text-align: center;
    margin: 0 auto;
    padding: 0;
}

.controls button:nth-child(1) {
    border: 2px solid #D2002E;
}

.controls button:nth-child(1) svg {
    color: #D2002E;
}

.controls button:nth-child(2) {
    border: 2px solid #008496;
}

.controls button:nth-child(2) svg {
    color: #008496;
}

.controls button:nth-child(3) {
    border: 2px solid #00B541;
}

.controls button:nth-child(3) svg {
    color: #00B541;
}

.controls > button {
    width: 45px;
    height: 45px;
    text-align: center;
    border-radius: 100%;
    margin: 0 6px;
    background: transparent;
}

.controls > button:hover svg {
    color: white;
}
</style>

<script>
// Get the modal
var modal = document.getElementById('id01');

```

```
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
feather.replace();

const controls = document.querySelector('.controls');
const cameraOptions = document.querySelector('.video-options>select');
const video = document.querySelector('video');
const canvas = document.querySelector('canvas');
const screenshotImage = document.querySelector('img');
const buttons = [...controls.querySelectorAll('button')];
let streamStarted = false;

const [play, pause, screenshot] = buttons;

const constraints = {
  video: {
    width: {
      min: 1280,
      ideal: 1920,
      max: 2560,
    },
    height: {
      min: 720,
      ideal: 1080,
      max: 1440
    },
  }
};
</script>
<script>
const getCameraSelection = async () => {
  const devices = await navigator.mediaDevices.enumerateDevices();
  const videoDevices = devices.filter(device => device.kind === 'videoinput');
  const options = videoDevices.map(videoDevice => {
    return `<option value="${videoDevice.deviceId}">${videoDevice.label}</option>
  });
  cameraOptions.innerHTML = options.join('');
};

</script>
<script>

play.onclick = () => {
  if (streamStarted) {
    video.play();
    play.classList.add('d-none');
    pause.classList.remove('d-none');
    return;
  }
  if ('mediaDevices' in navigator && navigator.mediaDevices.getUserMedia) {
    const updatedConstraints = {
      ...constraints,
      deviceId: {
        exact: cameraOptions.value
      }
    };
    startStream(updatedConstraints);
  }
};

const startStream = async (constraints) => {
  const stream = await navigator.mediaDevices.getUserMedia(constraints);
  handleStream(stream);
};
```

```
397
398  const handleStream = (stream) => {
399    video.srcObject = stream;
400    play.classList.add('d-none');
401    pause.classList.remove('d-none');
402    screenshot.classList.remove('d-none');
403    streamStarted = true;
404  };
405
406  getCameraSelection();
407
408  cameraOptions.onchange = () => {
409    const updatedConstraints = {
410      ...constraints,
411      deviceId: {
412        exact: cameraOptions.value
413      }
414    };
415    startStream(updatedConstraints);
416  };
417
418  const pauseStream = () => {
419    video.pause();
420    play.classList.remove('d-none');
421    pause.classList.add('d-none');
422  };
423
424  const doScreenshot = () => {
425    canvas.width = video.videoWidth;
426    canvas.height = video.videoHeight;
427    canvas.getContext('2d').drawImage(video, 0, 0);
428    screenshotImage.src = canvas.toDataURL('image/webp');
429    screenshotImage.classList.remove('d-none');
430  };
431
432  pause.onclick = pauseStream;
433  screenshot.onclick = doScreenshot;
434  </script>
435
436  </body>
437  </html>
```

8. RESULTS:

The proposed procedure was implemented and tested with set of images. The set of 15750 images of Alphabets from “A” to “I” are used for training database and a set of 2250 images of Alphabets from “A” to “I” are used for testing database. Once the gesture is recognise the equivalent Alphabet is shown on the screen.



9. ADVANTAGES AND DISADVANTAGES:

Advantages:

- Artificial intelligence (AI)powered smartphone app for deaf and mute people, which it says offers a low cost and superior approach to translating sign language into text and speech in real-time.
- AI technology solutions enable people with disabilities to gain more autonomy and be comfortable on their own.
- As different sign language standards exist, their dataset can be added, and the user can choose which sign language to read.

Disadvantages:

- Speech interpretation is helpful for sign language non-speakers who want the accompanying hand sign to be understood. Room conditions such as
- lighting can play a role in predicting the outcome of poor lighting.

- The light that is either too bright or too dim will result in inaccurate hand segmentation, resulting in inaccurate gesture prediction.
- As the quantity/quality of images in the dataset is low, the accuracy is not great,but that can easily be improved by change in dataset.

10. CONCLUSION:

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

11. FUTURE SCOPE:

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

12.APPENDIX:

Source code

Camera **:**

32 lines (31 sloc) | 1.12 KB

Raw Blame

```python
import cv2
import numpy as np
from keras.models import load_model
from keras.utils import load_img, img_to_array

class Video(object):
  def __init__(self):
    self.video = cv2.VideoCapture(0)
    self.roi_start = (50, 150)
    self.roi_end = (250, 350)
    self.model = load_model('G5F.h5')
        # Execute Local Trained Model
    self.model = load_model('G5F.h5') # Execute IBM Trained Model
    self.index=['A','B','C','D','E','F','G','H','I']
    self.y = None
  def __del__(self):
    self.video.release()
  def get_frame(self):
    ret,frame = self.video.read()
    frame = cv2.resize(frame, (640, 480))
    copy = frame.copy()
    copy = copy[150:150+200,50:50+200]
                # Prediction Start
    cv2.imwrite('image.jpg',copy)
    copy_img = load_img('image.jpg', target_size=(64,64))
    x = img_to_array(copy_img)
    x = np.expand_dims(x, axis=0)
    pred = np.argmax(self.model.predict(x), axis=1)
    self.y = pred[0]
    cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0)
    ret,jpg = cv2.imencode('.jpg', frame)
    return jpg.tobytes()
```

Main:

13 lines (10 sloc) | 207 Bytes

```
1   import cv2
2
3   video = cv2.VideoCapture(0)
4
5   while True:
6           ret, frame = video.read()
7           cv2.imshow("Frame", frame)
8           k = cv2.waitKey(1)
9           if k == ord('q'):
10                  break
11
12  video.release()
13  cv2.destroyAllWindows()
```

Mainstream :

23 lines (18 sloc) | 531 Bytes Raw Bla

```python
from flask import Flask, Response, render_template
from camera import Video

app = Flask(__name__)
@app.route('/')
def index():
        return render_template('Html_page_code.html')

def gen(camera):
        while True:
                frame = camera.get_frame()
                yield(b'--frame\r\n'
                        b'Content-Type: image/jpeg\r\n\r\n' + frame +
                        b'\r\n\r\n')

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


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

DEMO LINK:

https://drive.google.com/file/d/1Ev5U5hEElK5OYd_A4y7lM1PJxoyEtP9B/view?usp=drivesdk

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-32883-1660212854