

PROJECT REPORT

1.INTRODUCTION

1.1 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. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

1.2 Purpose

The 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. 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.

2.LITERATURE SURVEY

2.1 Existing Problem

Artificial intelligence is not designed to replace humans but rather to enhance our lives by helping us do things we are unable to do on our own. Many companies are working on this type of research, including Google Deep mind, IBM Watson, Apple Siri, Microsoft Cortana, etc., which means there will likely be many new developments soon. These innovations could positively impact everyone's life – even those without disabilities – because they make everyday tasks easier and less time-consuming.

2.2 References

1) A Signer Independent Sign Language Recognition with Coarticulation Elimination from Live Videos: an Indian Scenario P.K. Athira, C.J. Sruthi, A. Lijiya (2019)

Advantage: Economical can be implemented with a mobile camera which makes it very user-friendly.

Disadvantage: Not efficient under cluttered backgrounds and different illumination conditions.

2) White, J.J.: Fairness of AI for people with disabilities: problem analysis and interdisciplinary collaboration. ACM SIGACCESS Access. Comput. 125, 1 (2020)

Much has been written about the potential of artificial intelligence (AI) to support, and even transform, the lives of disabled people. It is true that many advances have been made, ranging from robotic arms and other prosthetic limbs supported by AI, decision support tools to aid clinicians and the disabled themselves, and route planning software for those with visual impairment. Many individuals are benefiting from the use of such tools, improving our accessibility and changing lives. But what are the true limits of such tools? What are the ethics of allowing AI tools to suggest different courses of action, or aid in decision-making? And does AI offer too much promise for individuals? I have recently undergone a life changing accident which has left me severely disabled, and together with my daughter who is blind, we shall explore the day-to-day realities of how AI can support, and frustrate, disabled people. From this, we will draw some conclusions as to how AI software and technology might best be developed in the future.

3) A Deep Learning based Indian Sign Language Recognition System Sruthi C. J and Lijiya A (2019)

Advantage: Training accuracy of 99.93% and with testing and validation accuracy of 98.64%.
Disadvantage: Facial expression and context analysis are the other part not included.

4) Bigham, J. P., Jayant, C., Miller, A., White, B., & Yeh, T. (2010, June). VizWiz::Locate It-enabling blind people to locate objects in their environment. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops (pp. 65-72). IEEE.

The sixth sense is a multi-platform app for aiding the people in need that is people who are handicapped in the form of lack of speech (dumb), lack of hearing (deaf), lack of sight (blind), lack of judicial power to differentiate between objects (visual agnosia) and people suffering from autism (characterized by great difficulty in communicating and forming relationships with other people and in using language and abstract concepts). Our current implementation of the product is on two platforms, namely, mobile and a web app. The mobile app even works for object detection cases in offline mode. What we want to achieve using this is to make a better world for the people suffering from disabilities as well as an educational end for people with cognitive disabilities using our app. The current implementation deals with object recognition and text to speech and a speech to text converter. The speech to text converter and text to speech converter utilized the Web Speech API (Application Program Interface) for the website and text to speech and speech to text library for the mobile platform. The object recognition wouldn't fetch enough use out of a website. Hence, it has been implemented on the mobile app utilizing the Firebase ML tool kit and different pre-trained models, which are both available offline as well as online.

2.3 Problem Statement Definition

Problem Statement (PS):

People with disabilities are not able to communicate with the people and society. Though technologies are evolving but there is no significant growth for these people. So, an AI system is developed to communicate with people in real time.

I am:

(Specially abled person)

A Specially abled person, who finds difficulties in communicating with the people and couldn't able to convey what they feel. And so, the talented ones not able to express what they feel.

I'm trying to:

(communicate with people)

Communicate with normal persons to convey the information which I intend to.

But:

(people find it difficult to understand)

I can't able to communicate easily with the people and they find it difficult to understand.

Because:

(they don't understand hand signs)

Only few knows the hand sign language not most of the people knows. So, it is a problem that every impaired person has.

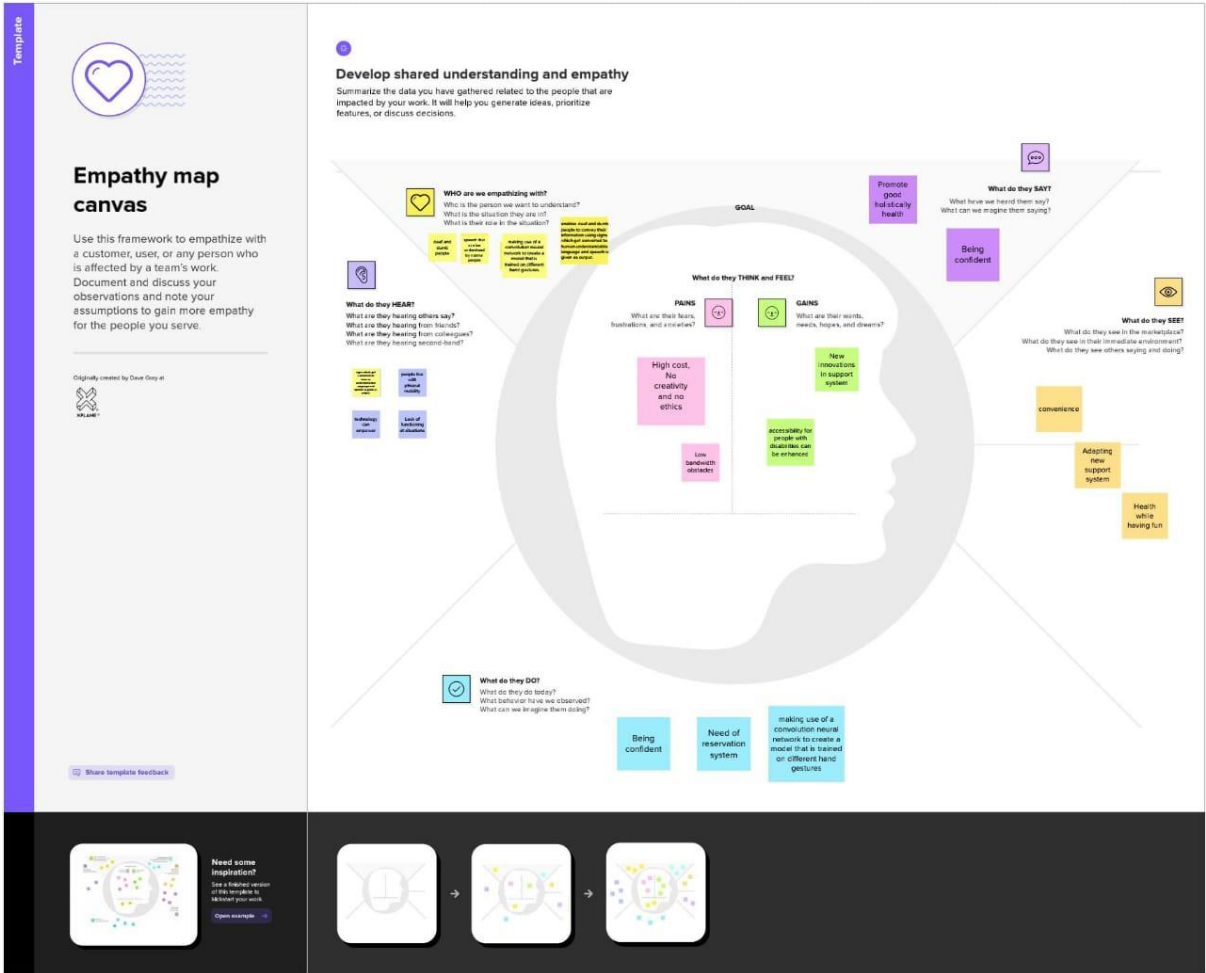
Which makes me feel:

(Anxiety, Lose confidence)

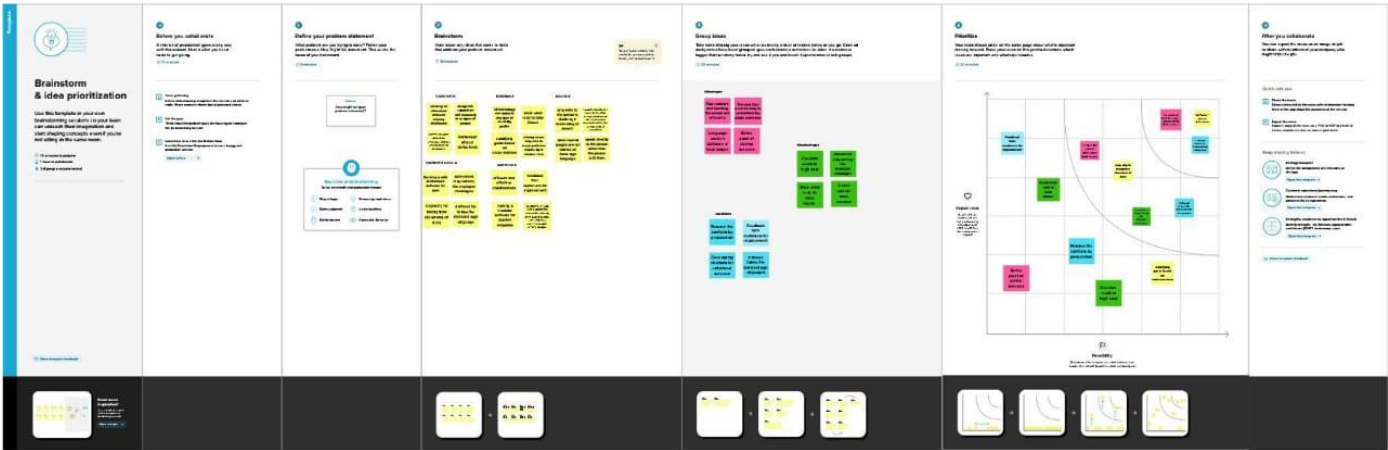
Frustrated, Lose confidence, Anxiety.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	An application for deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech in Artificial Intelligence
2.	Idea / Solution description	By using Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation
3.	Novelty / Uniqueness	We are using a convolution neural network to create a model that is trained on different hand gestures and an app is built for the use this mode
4.	Social Impact / Customer Satisfaction	Communicating with others and being connected in the society and remove accessibility barriers
5.	Business Model (Revenue Model)	By Using: Better communication with the disabled and Financial By Without Using: Can't Communicate and leads to loneliness
6.	Scalability of the Solution	Enhance people with disabilities to step into a world where they are facing difficulties in communication

3.4 Problem – SolutionFit

The Problem-Solution Fits simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem . It helps entrepreneurs , marketers and corporate innovators identify behavioral patterns and recognize what would work and why

Purpose:

- Solve complex problems in a way that fits the state of your customers.
- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- Sharpen your communication and marketing strategy with the right triggers and messaging.
- Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
- Understand the existing situation in order to improve it for your target group.

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? I.e. working parents of 0-5 y.o. kids The deaf and dumb, whom we collectively term as the “Specially-abled” 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. The specially-abled people find difficulties in communication with others. This makes them reluctant to encounter new environment and people.	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 Deaf and dumb tend to write or text in order to communicate which is found unviable in absence of necessary materials. They also make use of lip-reading, gestures and pointers to communicate.	Explore AS, differentiate
	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. Conversion of sign language into audio and text messages.	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. Normal people don’t take any effort to learn sign language which makes the communication with the specially-abled difficult.	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) They seek for interpreters and mobile applications to build communication with normal people.	
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. The ease of communication by the normal people.	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. To develop a web-based application to facilitate the communication between the normal and the specially-abled people using advanced deep learning algorithm.	8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 Social media application like Twitter, WhatsApp etc. 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. Local Community Camps conducted by NGOs, advertorial posters and interpreters.	Extract online & offline CH of BE
	4. EMOTIONS: BEFORE / AFTER EM Before: Feeling unfair about their communication ability when compared to normal people. After: Feeling better and bridging the gaps between people.			

4 .REQUIREMENT ANALYSIS

4.1 Functional Requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	System	Desktop with high resolution camera.Provides Access to capture Image through the Camera. Provides Access to Upload the Captured image through Gallery.
FR-4	Text conversion	Converts the Sign language into a textusing Convolutional Neural Network (CNN) Model.
FR-5	Sentence Translation	Recognizes the separate Signs of One-By-One and it Could provide aTranslation in the situation where Signed Extract System (SEE) is provided.
FR-6	Review	Users Can Give their Feedback on theReview page about the Application.

4.2 Non-Functional Requirement

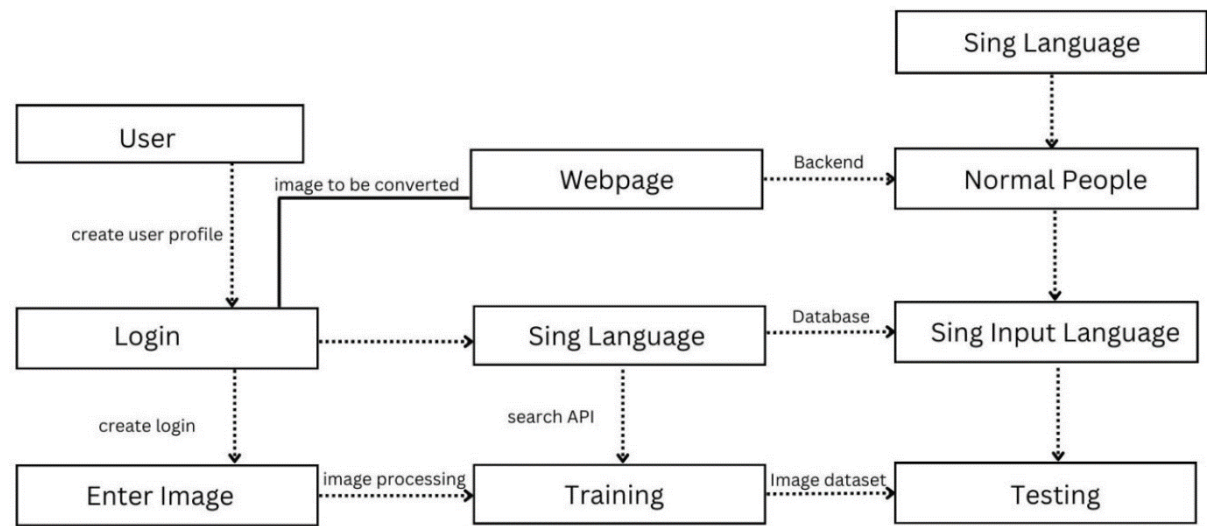
Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	To convey a message to normal people,as well as convert speech into understandable sign language for the deaf and dumb people.
NFR-2	Security	Converted information using signs into speech is accessed only by the user.
NFR-3	Reliability	Sign Method is Relevant to use for Differently abled persons.
NFR-4	Performance	The time for converting signs into speech should be faster for the real timecommunication.
NFR-5	Availability	Provides automatic recovery as much as possible.
NFR-6	Scalability	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.

5.PROJECT DESIGN

5.1 Data Flow Diagrams

A data flow diagrams (DFD) is a traditional visual representation of the information flows with in a system. A clear DFD can depict the right amount of the system requirement graphically. It shows data enter and leaves the system. What changes the information.



5.2 Solution & Technical Architecture:

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

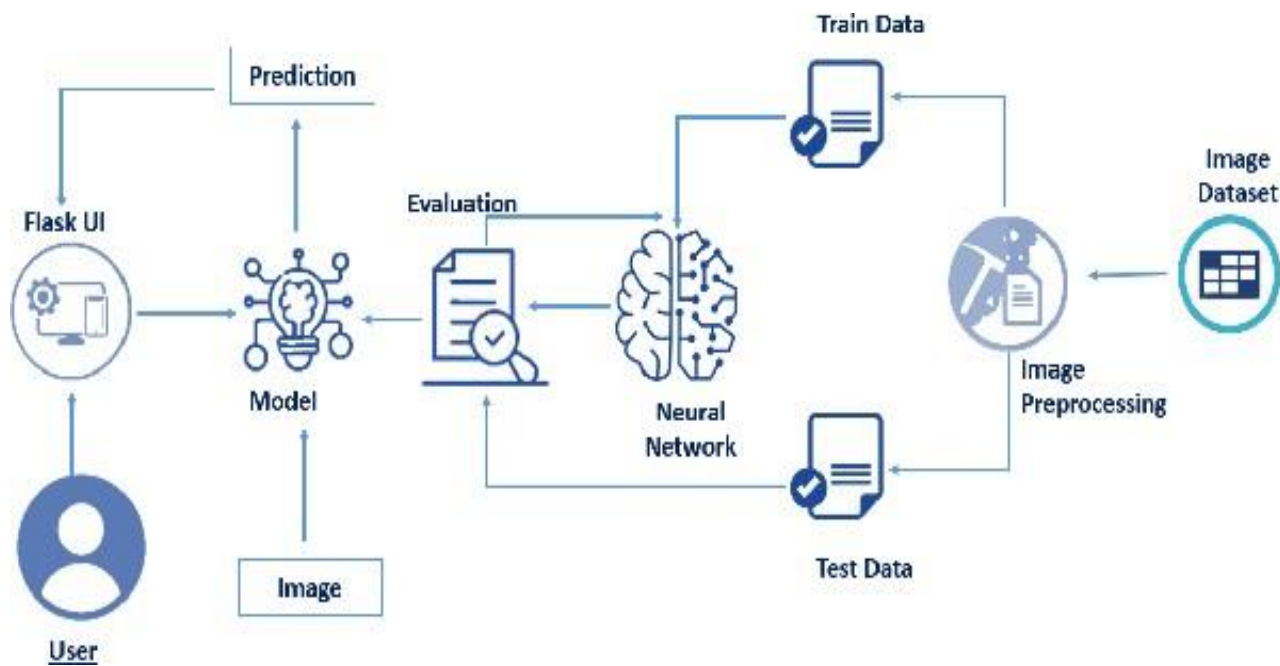


Table: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Chat bot user interface	HTML, CSS, Python.
2.	Application Logic	Logic for a process in the application	Python
3.	Application Logic	Logic for a process in the application	IBM Watson STT service & TTS service
4.	Cloud Database	Database Service on Cloud	IBM cloudant
5.	File Storage	File storage requirements	Local File system
6.	Machine Learning Model	Neural Networks –CNN model, ANN model	Object Recognition Model –CNN model
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System	Local, Cloud Foundry, Kubernetes
8.	External Interfaces	Any interface that is transmitting information from the product to a third-party may contain information that is useful for an attack	Operating System - Windows, Mac, Linux; CPU & GPU (for training), WebCam Scanners, Speakers and PC

5.3 User stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance Criteria	Priority	Release
Customer (Low Vision)	Registration	USN-1	As a user, who has trouble reading due to low vision, I want to be able to market the text larger on the screen so that I can read it.	I can access my account/dash board	High	Sprint-1
Customer (Color blindness)		USN-2	As a user, who is color blind, I want to have access to information conveyed in color so that I do not miss anything and I understand the content.	I can receive confirmation email and click confirm	High	Sprint-1
Customer (Impaired User)		USN-3	As a user, who is hearing - paired, i want a transcript of the spoken audio so that I can have access to all information provided in audio clips	I can register and access the dashboard with facebook login	High	Sprint-2

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Milestone	Function (Epic)	Milestone Story Number	Story / Task
Milestone 1	Data Collection	M1	We're collecting dataset for building our project and creating to folders, one for training and another one for testing
Milestone 2	Image Processing	M2	Importing image data generator libraries and applying image data generator functionally to train the test set.
Milestone 3	Building Model	M3	Importing the model building libraries, Insulation the model, Adding Convolution layers, Adding the Polling layers, Adding the Flatten layers, Adding Dense layer, Compiling the model Fit and Save the model.
Milestone 4	Testing Model	M4	Import the packages first. Then we save the model and Load the test image. Pre-progress it and predict it.
Milestone 5	Application layer	M5	Build the flask application and the HTML pages.
Milestone 6	Train Conversation Engine	M6	Register for IBM cloud and train image classification mode
Milestone 7	Final Result	M7	To ensure all the activities and resulting the final output.

6.2 Sprint Delivery Schedule

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Dataset is collected on the basis of various hand signs and curated according to the problem statement.	4	High	Nachiyar , Rakshini,Santhiya,Naveetha banu,Kaviya
Sprint-1	Data Preprocessing	USN-2	The dataset is preprocessed in order to check noisy data and other inconsistencies before executing it to the algorithm	6	Medium	Nachiyar,Rakshini,Santhiya,
Sprint-2	Model Building	USN-3	Model is built according to the image features in such a way that the model identifies the features of the sign image and learns in order to give correct output.	8	High	Nachiyar,Rakshini,Santhiya, Naveetha banu,Kaviya

Sprint-3	Model Training	USN-4	Data is fed into the model and the model is trained in order to find the optimal weights that give help in predicting the correct output.	8	Medium	Santhiya,Naveetha banu,Kaviya
Sprint-3	Training and Testing	USN-5	Model is tested in such a way that the collection data or images are trained frame by frame according to the user requirements.	6	High	Nachiyar,Rakshnini
Sprint-4	Implementation of the application	USN-6	Converting the input sign language images into English alphabets	8	Medium	Nachiyar,Rakshini,Santhiya,Naveetha banu,Kaviya

Project Tracker, Velocity & Burndown Chart:

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

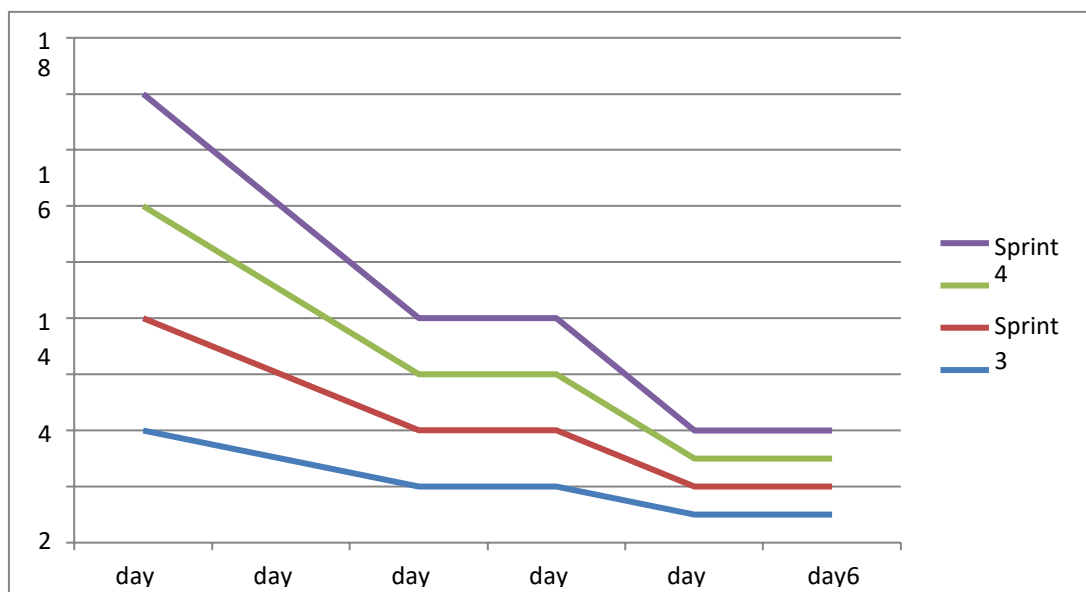
Velocity:

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)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



7.CODING & SOLUTIONING

7.1 Feature

The project is the flask application for converting hand signs to speech. Home page has a record button. By pressing the button, the hand signs will be read from the camera. In the backend, the classifier converts the hand sign to the respective character. The converted character is displayed on the screen. On pressing Esc button, the audio of the shown hand sign will be played.

The file `asl_classifier.h5` contains the model for converting hand signs to text. The model is loaded using the keras library, `keras.models.load_model`. `lables_dict` maps the model's output to the alphabets.

Python provides various libraries for image and video processing. One of them is OpenCV. OpenCV is a vast library that helps in providing various functions for image and video operations. With OpenCV, we can capture a video from the camera. It lets you create a video capture object which is helpful to capture videos through a webcam and then you may perform desired operations on that video.

Steps to capture a video:

Use `cv2.VideoCapture()` to get a video capture object for the camera.

Set up an infinite while loop and use the `read()` method to read the frames using the above created object. Use `cv2.imshow()` method to show the frames in the video.

Breaks the loop when the user clicks a specific key.

There are several APIs available to convert text to speech in Python. One of such APIs is the Google Text to Speech API commonly known as the gTTS API. gTTS is a very easy to use tool which converts the text entered, into audio which can be saved as a mp3 file. The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. gtts is used in the project to convert the text to speech.

8. TESTING

8.1 Test Cases

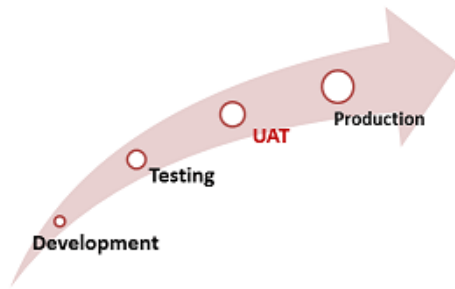
Test cases are the hand signs for each alphabet



8.2 User Acceptance Testing

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

Purpose of UAT



The main **Purpose of UAT** is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved.

Need of User Acceptance Testing

Need of User Acceptance Testing arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

- 1 {
 - Developers have included features on their "own" understanding
- 2 {
 - Requirements changes "not communicated" effectively to the developers

Developers code software based on requirements document which is their “own” understanding of the requirements and **may not actually be what the client needs from the software.**

Requirements changes during the course of the project may not be communicated effectively to the developers.

9. RESULTS

9.1 Performance Metrics

The accuracy of the model is 0.96 by training the with the epoch of 10.

```
model.fit_generator(x_train, steps_per_epoch=24, epochs=10, validation_data=x_test, validation_steps=40)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated
future version. Please use `Model.fit`, which supports generators.
  """Entry point for launching an IPython kernel.

Epoch 1/10
24/24 [=====] - 17s 666ms/step - loss: 1.9786 - accuracy: 0.5628
Epoch 2/10
24/24 [=====] - 16s 662ms/step - loss: 1.4525 - accuracy: 0.6621
Epoch 3/10
24/24 [=====] - 16s 676ms/step - loss: 0.9580 - accuracy: 0.6842
Epoch 4/10
24/24 [=====] - 16s 675ms/step - loss: 0.7076 - accuracy: 0.7240
Epoch 5/10
24/24 [=====] - 16s 659ms/step - loss: 0.6103 - accuracy: 0.7488
Epoch 6/10
24/24 [=====] - 16s 663ms/step - loss: 0.5005 - accuracy: 0.8054
Epoch 7/10
24/24 [=====] - 17s 679ms/step - loss: 0.4164 - accuracy: 0.8904
Epoch 8/10
24/24 [=====] - 18s 723ms/step - loss: 0.3408 - accuracy: 0.8994
Epoch 9/10
24/24 [=====] - 16s 659ms/step - loss: 0.2641 - accuracy: 0.9536
Epoch 10/10
24/24 [=====] - 16s 664ms/step - loss: 0.1676 - accuracy: 0.9672
```

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.**

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

It helps to listen to class notes, text books and electronic text.

- ➡ It facilitates education.
- ➡ It avoids eyestrain from too much reading.
- ➡ It helps in learning languages which you do not know.
- ➡ It helps in preparation of speeches by hearing your work read aloud.
- ➡ It helps in listening e-books or e-material during journey.
- ➡ It amuses children by letting your PC read stories to them when you are busy.
- ➡ It helps seniors or those having vision problems.
- ➡ It can be adapted easily to say whatever users want them to say.
- ➡ It can help in reading large paragraphs and offers range of different accents and voices.

10.2 Disadvantages

- ➡ The system is very time consuming as it requires huge databases and hard-coding of combination to form these words. As a result speech synthesis consumes more processing power.
- ➡ The resulting speech is less than natural and emotionless. This is because it is impossible to get audio recordings of all possible words spoken in all the possible combinations of emotions, prosody, stress etc.
- ➡ Pronunciation analysis from written text is a major concern.
- ➡ It is difficult to build a perfect system.
- ➡ Filtering background noise is a task which can even be difficult for humans to accomplish.

11. CONCLUSION

AI holds the key to unlocking a magnificent future where, driven by data and computers that understand our world, we will all make more informed decisions. These computers of the future will understand not just how to turn on the switches but why the switches need to be turned on.

The project aims at converting sign gestures into speech that can be understood by normal people. The entire model pipeline is developed by CNN architecture for the classification of 26 alphabets and one extra alphabet for null character. The model has achieved an efficiency of 96.08% .

12. FUTURE SCOPE

Model enhancement could bring better accurate values. It takes nearly 3 seconds for the model to predict the hand signs. A better model could be used to predict the hand signs in few milliseconds.

We are making use of a **convolution neural network to create a model that is trained on different hand gestures.**

13. APPENDIX

Source Code:

```
!pip install keras
!pip install tensorflow
!pip install opencv-python
import cv2
from keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
# from keras.preprocessing.image import load_img, img_to_array
import numpy as np
import tensorflow as tf
import keras
model = keras.models.load_model("asl_classifier.h5")
labels_dict = {0:'0',
               1:'A',
               2:'B',
```

3:'C',
4:'D',
5:'E',
6:'F',
7:'G',
8:'H',
9:'I',
10:'J',
11:'K',
12:'L',
13:'M',
14:'N',
15:'O',
16:'P',
17:'Q',
18:'R',
19:'S',
20:'T',
21:'U',
22:'V',
23:'W',
24:'X',
25:'Y',
26:'Z'}

color_dict=(0,255,0)

x=0

y=0

w=64

h=64

Fully Real-Time:

```

img_size=128
minValue = 70
source=cv2.VideoCapture(0)
count = 0
string = " "
prev = " "
prev_val = 0
while(True):
    ret,img=source.read()
    gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    #cv2.rectangle(img,(x,y),(x+w,y+h),color_dict,2)
    cv2.rectangle(img,(24,24),(250 , 250),color_dict,2)
    crop_img=gray[24:250,24:250]
    count = count + 1
    if(count % 100 == 0):
        prev_val = count
cv2.putText(img,str(prev_val//100),(300,150),cv2.FONT_HERSHEY_SIMPLEX,1.5,(
255,255,255),2)
blur = cv2.GaussianBlur(crop_img,(5,5),2)
th3=cv2.adaptiveThreshold(blur,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.T
HRESH_BINARY_INV,11,2)
ret,res=cv2.threshold(th3,minValue,255,cv2.THRESH_BINARY_INV+cv2.THRESH_
OTSU)
resized=cv2.resize(res,(img_size,img_size))
normalized=resized/255.0
reshaped=np.reshape(normalized,(1,img_size,img_size,1))
result = model.predict(reshaped)
#print(result)
label=np.argmax(result,axis=1)[0]
if(count == 300):
    count = 99
    prev= labels_dict[label]

```

```

if(label == 0):
    string = string + " "
    #if(len(string)==1 or string[len(string)] != " "):

else:
    string = string + prev
cv2.putText(img,prev,(24,14),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)
cv2.putText(img,string,(275,50),cv2.FONT_HERSHEY_SIMPLEX,0.8,(200,200,200),
2)

cv2.imshow("Gray",res)
cv2.imshow('LIVE',img)
key=cv2.waitKey(1)

if(key==27):#press Esc. to exit
    break
print(string)
cv2.destroyAllWindows()
source.release()

cv2.destroyAllWindows()
!pip install gTTS
!pip3 install --upgrade setuptools
!pip3 install playsound pip install playsound
from gtts import gTTS
import playsound

# This module is imported so that we can
# play the converted audio
import os

```



```
# The text that you want to convert to audio

# Language in which you want to convert
language = 'en'

# Passing the text and language to the engine,
# here we have marked slow=False. Which tells
# the module that the converted audio should
# have a high speed
myobj = gTTS(text=string, lang=language, slow=False)

# Saving the converted audio in a mp3 file named
# welcome
myobj.save("welcome.mp3")

# Playing the converted file
playsound.playsound("welcome.mp3")
from playsound import playsound
playsound('welcome.mp3')
```

GitHub Link :

<https://github.com/IBM-EPBL/IBM-Project-11847-1659348514>

Project Demo Link:

<https://youtu.be/WjzRJO48AzA>