

Project Report

Team ID	PNT2022TMID09233
Project Name	Real-Time Communication System Powered by AI for Specially Abled
Team Lead	Adarsh Srinivas (Roll No. 310619104003)
Team Member 1	Abubucker Siddique (Roll No. 310619104001)
Team Member 2	Ashwath (Roll No. 310619104012)
Team Member 3	Jayasurya (Roll No. 310619104044)

1. INTRODUCTION

1.1. PROJECT OVERVIEW

The project developed is a system that converts hand gestures of a Deaf-Mute individual into its respective ASL (American Sign Language) alphabets for a normal individual for communication. The main customer for our project are: People who want to communicate with deaf-mute individual who desire to communicate with others, and deaf-mute individual who desire to communicate with others. This project tries to solve the communication during the time of emergencies. The project is developed on Python Platform using CNN (Convolutional Neural Network) model from TensorFlow package.

1.2. PURPOSE

Everybody cannot afford to have a human translators of sign language, they may not be available all the time and they are quite expensive. People who engage in conversation with deaf-mute individual will find it hard and tedious. Deaf-mute individual may lose a lot of opportunities because they cannot speak or express their thoughts verbally in situations like an interview. This project aims to overcome the said challenges.

2. LITERATURE SURVEY

2.1. EXISTING PROBLEMS

- Existing system (or) frameworks has too many false positives. The system predicts the gestures inaccurately.
- Real Time recognition of gestures into text/speech and text/speech into gestures is not available.

2.2. REFERENCES

- [1] Saed Mian Qaisar, Sarah Niyazi, Abdulhamit Subasi, "Efficient Isolated Speech to Sign Conversion Based on the Adaptive Rate Processing"; Procedia Computer Science, Vol. 163, PP. 35–40, 2019.
- [2] T. Bohra, S. Sompura, K. Parekh and P. Raut, "Real-Time Two Way Communication System for Speech and Hearing Impaired Using Computer Vision and Deep Learning" International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 734-739, 2019.
- [3] Ma, Jiyong, Wen Gao, Jiangqin Wu, and Chunli Wang. "A continuous Chinese Sign Language recognition system." In Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), pp. 428-433. IEEE, 2000.
- [4] Vogler, C., and D. Handshapes Metaxas. "Movements: Multiple-Channel American Sign Language Recognition." Gesture-Based Communication in Human-Computer Interaction. Lecture Notes in Computer Science: 247-258.

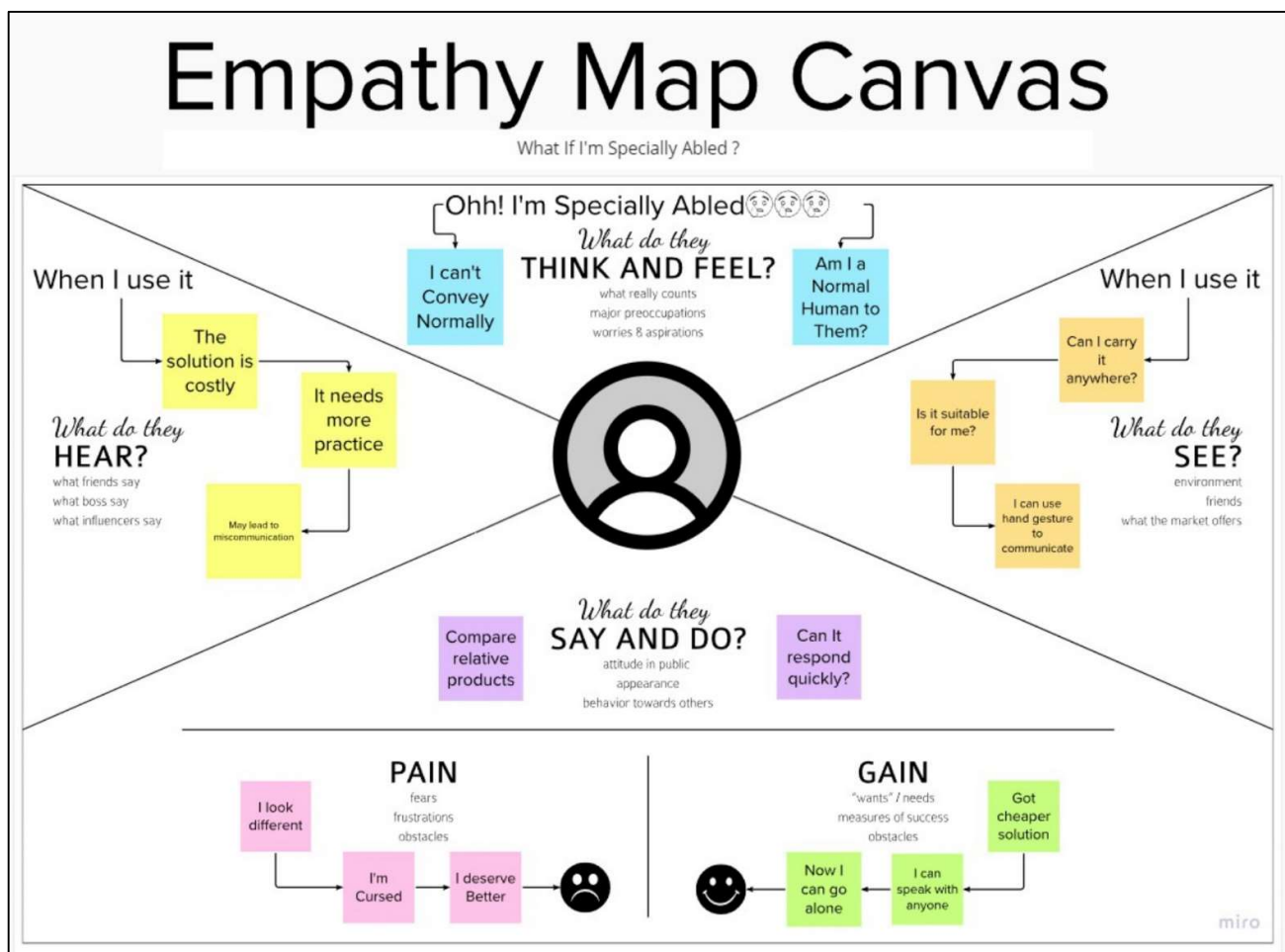
[5] Pavlovic, V, Sharma, R., &Huang T., “Visual Interpretation of Hand Gestures for Human-Computer Interaction (HCI): A Review”, IEEE TOPAMI, VOL. 19, NO. 7, 1999.

2.3. PROBLEM STATEMENT DEFINITION

The study of human-computer interaction has shown a great deal of interest in hand gesture recognition. In many areas of human-computer interaction, including virtual reality, gaming, automobile system control, and robotic control, quick and precise hand gesture recognition is crucial. As more sensors are added, there are numerous different ways to categorise hand motions. Since gesture identification is a problem of image classification and 2D CNNs are effective in image classification, we have chosen to employ a convolutional neural network for this task. A system that converts the sign language into the respective ASL (American Sign Language) alphabet to convey a message to normal people is developed in this project.


3. IDEATION & PROPOSED SOLUTION

3.1. EMPATHY MAP CANVAS



3.2. IDEATION & BRAINSTORMING

- Step-1: Team Gathering, Collaboration and Select the Problem Statement



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

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

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

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

[Open article](#) →

1 Problem Statement

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.

PROBLEM

How might we[your problem statement]?

Key rules of brainstorming

To run an smooth and productive sessi.

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

• Step-2: Brainstorm, Idea Listing and Grouping

2 Brainstorm

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

⌚ 10 minutes

Adarsh Srinivas

- Consider providing alternative
- Meeting to connect to the user only when it is necessary
- Using more gestures to convey a message
- Reducing time taken to connect
- Provide context explanation
- Use of convolution neural network
- Building confidence
- Efficiency in using the application

Ashwath

- Ensure user interface is optimized
- Clarity of the speech
- An overview of a video
- Home automation
- Take care of customer feedback
- Adding fun with work and work
- Secure 256/7 support
- Make the user feel comfortable
- Ensuring the safety of the actions

Abubucker Siddique

- Interacting with the user
- Ensuring the user is happy
- Using of good body language
- Ensure the user is happy
- Checking for improper sign
- Ensure for any error
- Interacting with the user
- Ensuring the user is happy
- Using of good body language
- Ensure the user is happy
- Checking for improper sign
- Ensure for any error

Jayasurya

- Users can click and select
- Managing the time
- Ensuring the user is happy
- Reducing the time
- Ensuring the user is happy
- Ensuring the user is happy
- Ensuring the user is happy
- Ensuring the user is happy

TIP
You can select a sticky note and hit the pencil (switch to sketch) icon to start drawing!

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 can break it up into smaller sub-groups.

⌚ 20 minutes

Authentication

- Fast login and logout
- Secured application

Technology

- Using convolution neural network
- Use of gestures

Others

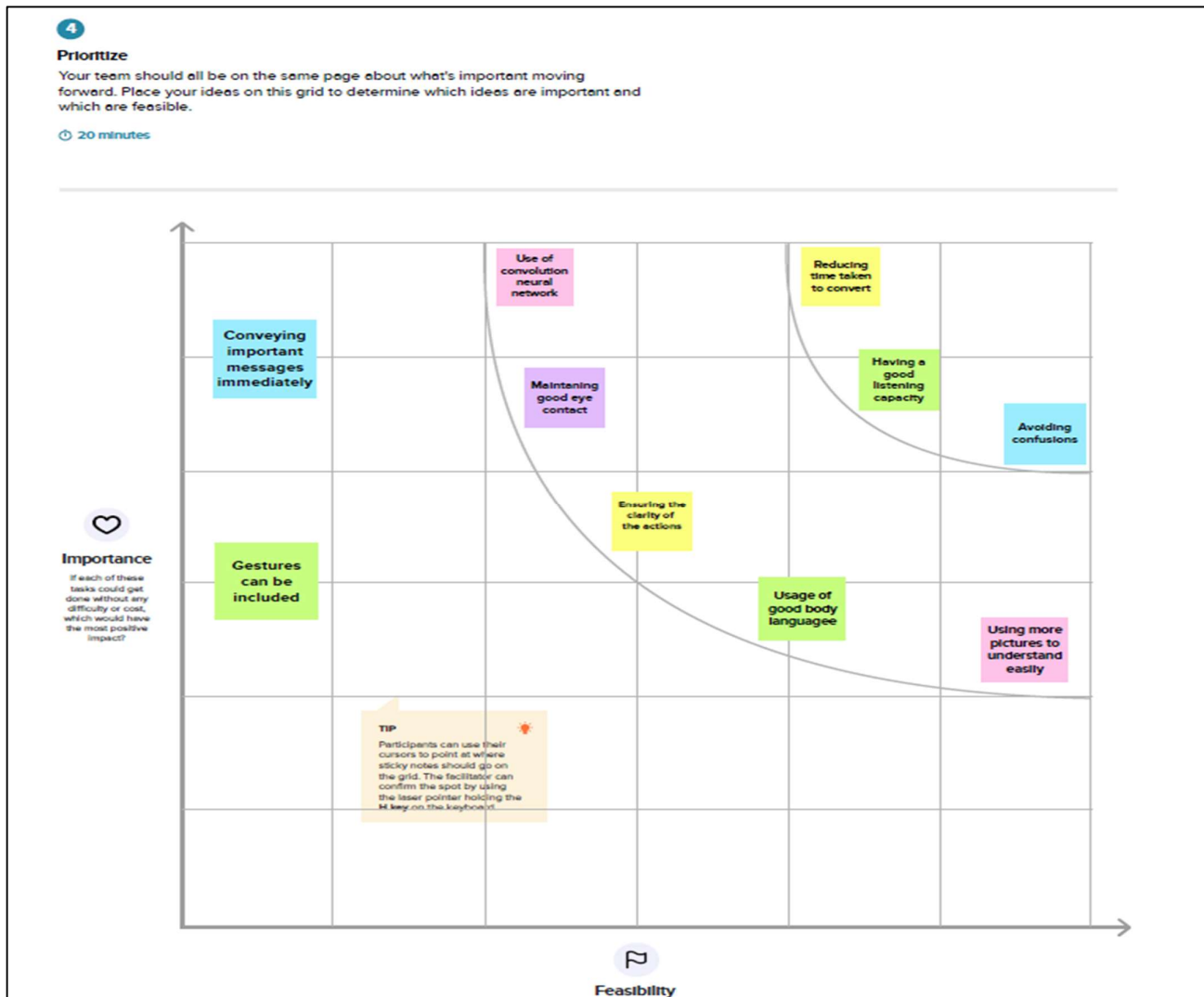
- Ensuring optimal user interface
- Working on the best interface while using
- Checking for improper signals
- Taking less time for conversion

Reviews

- Getting feedbacks from users
- Using the queries for betterment

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

• Step-3: Idea Prioritization



3.3. PROPOSED SOLUTION

S No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Correspondence between a speech and hearing-impaired individual and a normal person has always been a challenging task. It is very difficult for speech impaired individual to convey their message to normal people. Since normal people are not well-versed in Sign Language. Correspondence during the hour of crisis is a pain. Everyone can't stand to have human interpreters of Sign Language. Speech and hearing-impaired individuals might lose a ton of chances since they can't talk or offer their viewpoints verbally in circumstances like a meeting.
2.	Idea / Solution description	1. Real time detection and conversion of Sign to Speech/Text and Speech/Text to Sign System will be very useful to have a proper conversation between a normal person and an impaired person.

		<ol style="list-style-type: none"> 2. To design and implement a system using Artificial Intelligence, Convolutional Neural Network, Computer Vision and Image Processing to take hand gestures as input and convert them to Speech/Text.
3.	Novelty / Uniqueness	<ol style="list-style-type: none"> 1. An app called “VHearU” that uses Artificial Intelligence, computer vision, convolutional neural network and image processing. 2. It recognizes the image of sign language from the speaker and then translates it into speech/text.
4.	Social Impact / Customer Satisfaction	<ol style="list-style-type: none"> 1. Disabled people experience a great deal of difficulty with day-to-day activities. 2. The primary aim of this application is to increase the security, certainty and independence of speech-hearing impaired individual.
5.	Business Model (Revenue Model)	<ol style="list-style-type: none"> 1. AI can create income through direct customers and collaborate with health care sector and produce income from their customers. 2. Speech-hearing impaired employees of B2B services can use the app to pass messages concurring on to the organization.
6.	Scalability of the Solution	<ol style="list-style-type: none"> 1. AI innovation assists the specially abled with opening up new doors for availability consideration in the public eye and independent living. 2. It might also open the door to more cutting-edge and creative innovations to the most challenging problems faced by the specially abled.

3.4. PROBLEM SOLUTION FIT

Define CS, fit into CL	1. CUSTOMER SEGMENT(S) CS <p>The primary target audience for our project are: 1. Individuals who want to communicate with deaf and mute people. 2. Deaf and mute individuals who want to interact with others.</p>	6. CUSTOMER LIMITATIONS CL <small>EG. BUDGET, DEVICES</small> <p>Not every person grasps the gesture based communication. Thus, it becomes challenging for the impaired to communicate in everyday life. Communication through signing is a boon for the speech and hearing impaired individuals to offer their viewpoints and feelings.</p>	5. AVAILABLE SOLUTIONS AS <small>PROS & CONS</small> <p>The input for perceiving the signs given by the user. The significant inconvenience of the current framework is the intricacy of involving sensors for gesture recognition. The user is supposed to be wearing the gloves each time they need to give an input and it is very costly and challenging to utilize.</p>	Explore AS, differentiate
	2. PROBLEMS / PAINS + ITS FREQUENCY PR <p>1. Correspondence during the hour of crisis. 2. Everyone can't stand to have human interpreters of sign language. 3. Speech and hearing impaired individuals might lose a ton of chances since they can't talk or offer their viewpoints verbally in circumstances like a meeting.</p>	9. PROBLEM ROOT / CAUSE RC <p>At present, the innovative headways maneuver everybody into the computerized way of life. Subsequently, bringing about the issue of correspondence through the web, for the speech and hearing impaired individuals, who are comfortable just with the communication via gestures they know. Here comes the need to construct a gesture based communication recognition framework, through which the computer can be made to perceive and decipher gesture based communication and interpret it for the necessary task.</p>	7. BEHAVIOR + ITS INTENSITY BE <p>Directly related: D-Talk, communication through signing, hand motion, mental harm, trouble to communicate. Indirectly related: Enabled innovation, completely paralyzed, honorable goal, involving sensors in everyday life, ecological dangers influence their life trouble in the public eye.</p>	
Focus on PR, tap into BE, understand RC	3. TRIGGERS TO ACT TR <p>Speech and hearing impaired individuals can without much of a stretch fall into discouragement as they can only with significant effort express their sentiments. Likewise, their relatives go through a ton of stress to grasp them.</p>	10. YOUR SOLUTION SL <p>Individuals with disabilities stand to benefit fundamentally from AI-powered solutions, which will assist them with day to day errands and give them the opportunity to acquire new abilities. The project aims to foster a framework that changes over the gesture based communication into a human hearing voice in the ideal language to pass a message on to ordinary individuals, as well as convert speech into reasonable communication via gestures for the speech and hearing impaired individual.</p>	8. CHANNELS of BEHAVIOR CH <p>ONLINE Giving them specialised equipment to supplement educational programmes would help them become more mentally and emotionally healthy.</p>	Extract online & offline CH of BE
	4. EMOTIONS <small>BEFORE / AFTER</small> EM <p>Before: Weakness, Absence of Certainty, Dependent After Secured, Certainty, Independent</p>		<p>OFFLINE Their lives are mostly impacted by entering our direct society, which includes schools, colleges, and workplaces.</p>	
Identify strong TR & EM				

4. REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENT

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	Uploading image	Upload image through camera Upload image through gallery
FR-4	Text to speech	Select speech icon to convert the respective text for sign language
FR-5	Whiteboard	Use whiteboard to share the message by drawing

FR-6	Emergency templates	Select emergency templates icon to pass the message quickly
------	---------------------	---

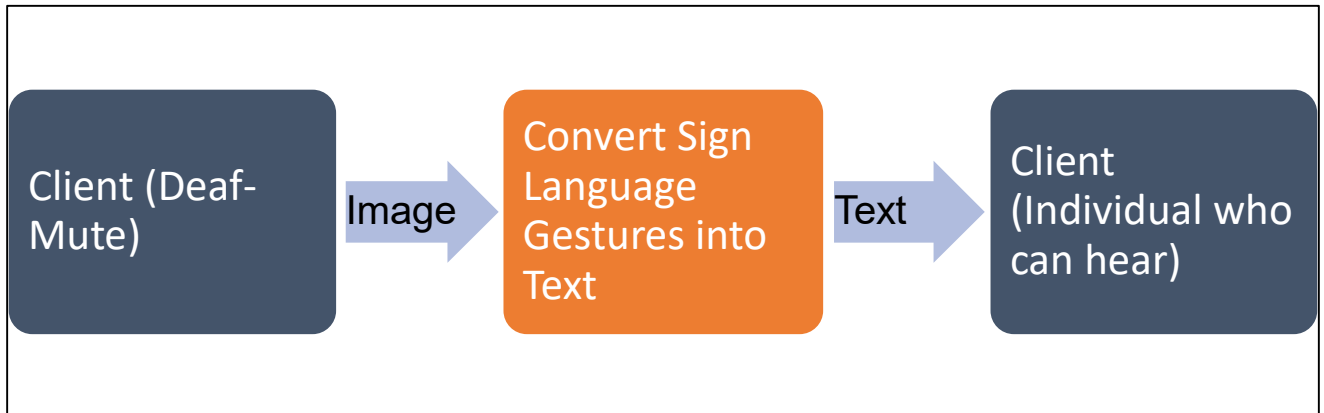
4.2. NON-FUNCTIONAL REQUIREMENT

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Client can undoubtedly upload the image and this application is planned in a manner here, client can without much of a stretch discover some predefined layouts
NFR-2	Security	Client should sign in into an app only then proceed for further process. So unapproved access will be kept away from at max.
NFR-3	Reliability	This application has robust adaptation to non-critical failure and regardless of whether an error happens likewise it recuperates rapidly.
NFR-4	Performance	This application will rapidly transfer and process the images since it predicts the gestures through signing utilizing CNN model and it gives high accuracy.
NFR-5	Availability	The predefined formats will be accessible to all clients and furthermore have whiteboard choice. This application is planned such that it is straightforward and accessible to all clients.
NFR-6	Scalability	Engineers can add new formats and it will build adaptability and this application has premium elements where client approach google maps and google duo.

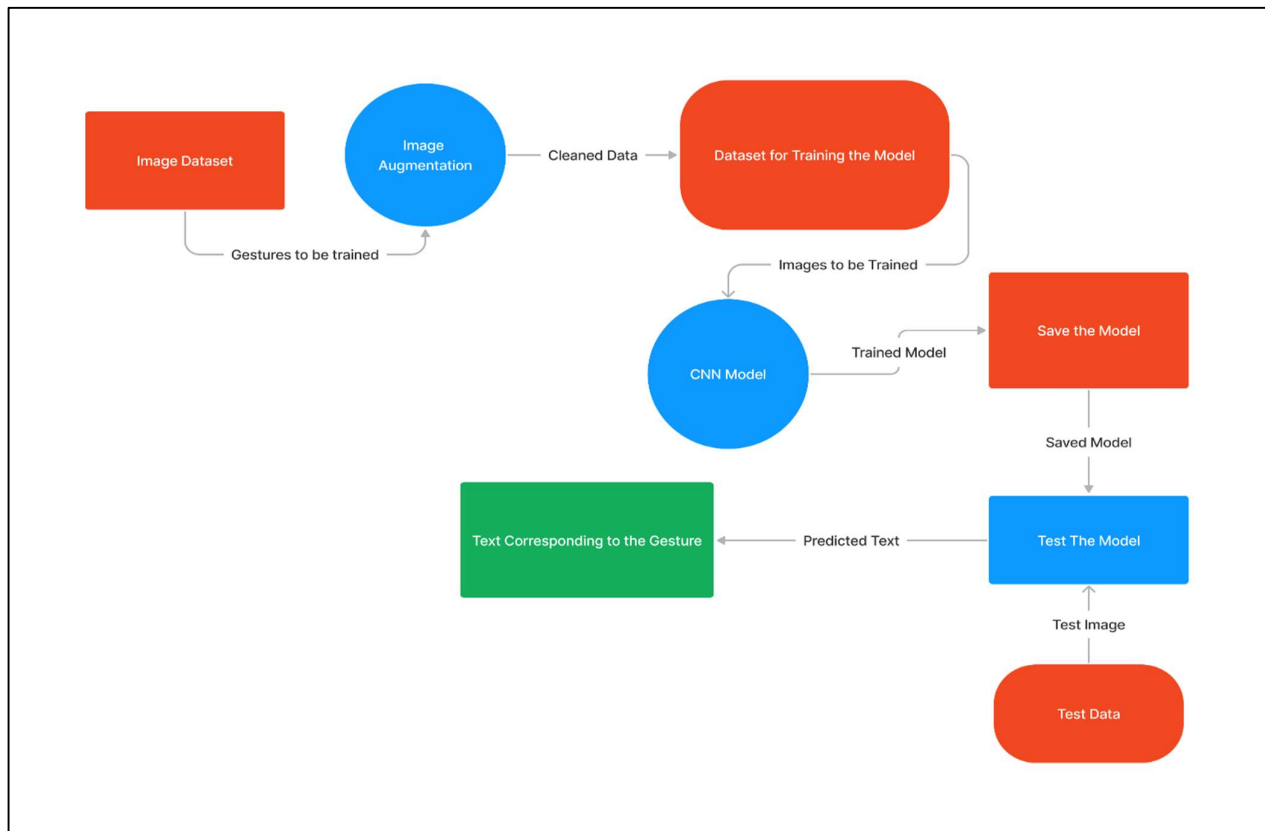
5. PROJECT DESIGN

5.1. DATA FLOW DIAGRAMS

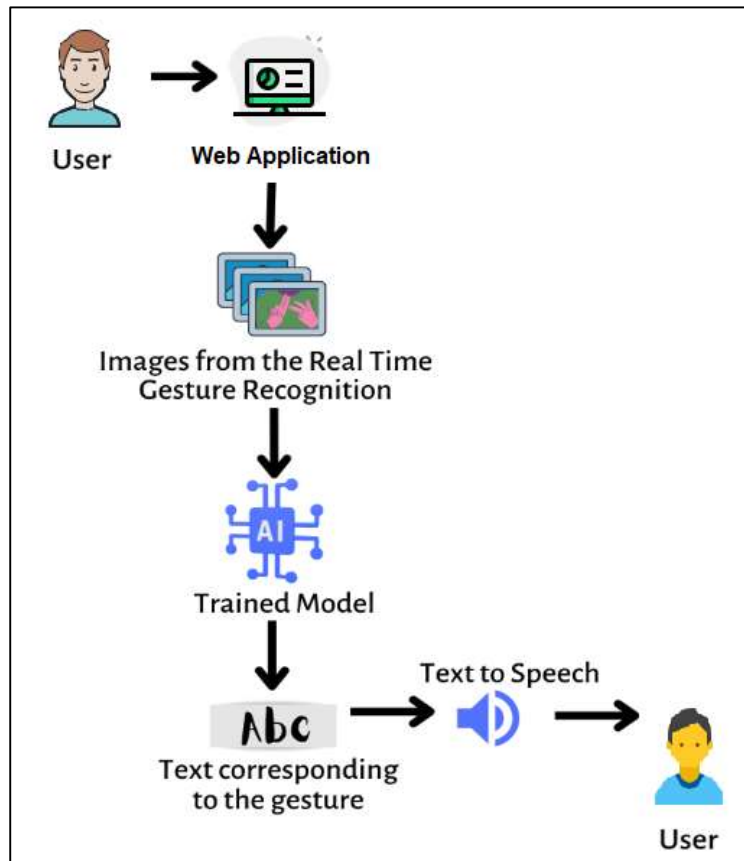
- Level 0 - DFD



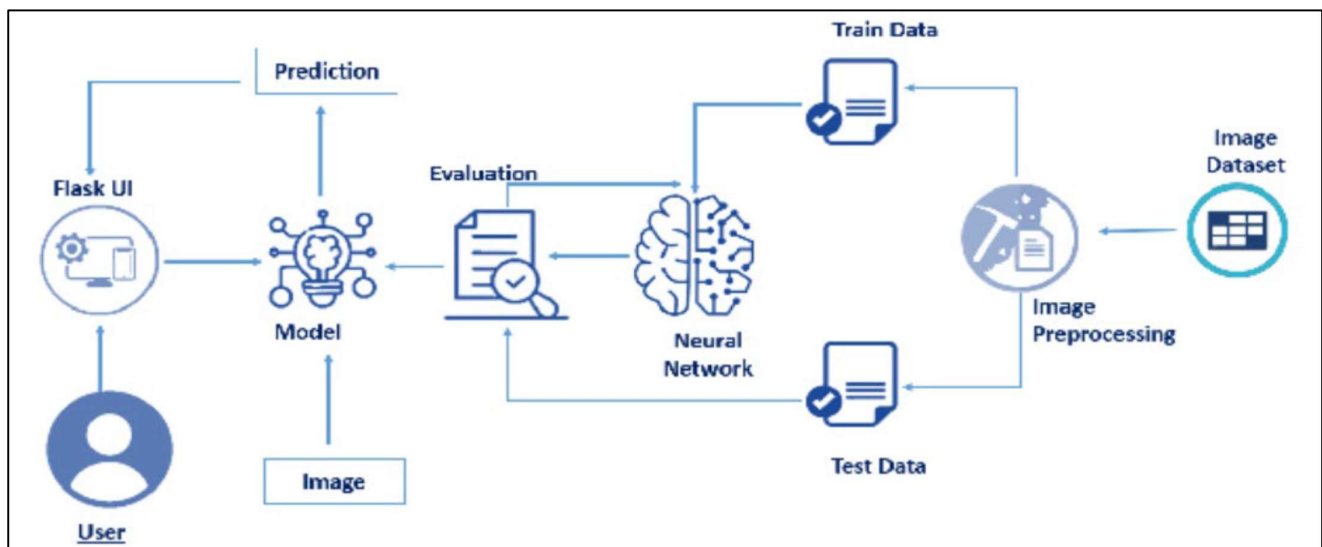
- Level 1 - DFD



- Simplified Flow Diagram



5.2. SOLUTION & TECHNICAL ARCHITECTURE



5.3. USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria
Customer (People who cannot hear)	Convert sign language into text	USN - 1	As a user, I can open camera in the app and record my signs to be converted into text	I can communicate with normal people effectively

		USN - 2	As a user, I can upload my previous sign gestures to communicate faster	I can have a list of frequently used signs to make for fast reference
	Dashboard	USN – 3	Buttons to record the signs, to convert in real-time and other buttons should be available in the right places	All features must be easily accessible
		USN – 4	Emergency calls must be available so that I can press a button in times of emergency to get the attention of others	I can feel safe because of the Emergency Button which can get me help.
Customer (People who can hear and talk)	Convert sign language into text	USN - 5	As a user, I can open back camera in the app and record the specially abled people's signs to be converted into text	I can understand the mode of communication of specially-abled people effectively
	Show the message to convey in the form of text	USN - 6	As a user, I can open a Text-pad that is available in the app, so that the deaf people can see the message I need to convey	I can convey my message to them effectively
Administrator	Integrate application with trained model	USN - 7	As an admin, I should be able to integrate the AI model into the application and maintain the application	I can give best experience to app users

6. PROJECT PLANNING & SCHEDULING

6.1. SPRINT PLANNING & ESTIMATION

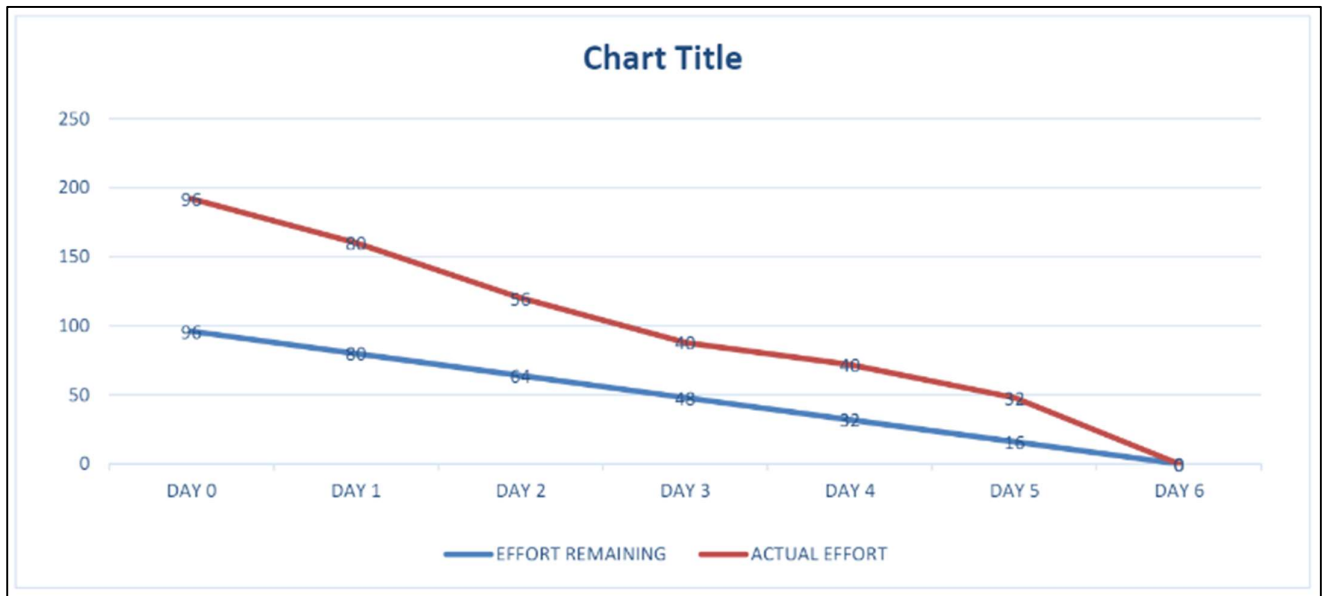
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
--------	-------------------------------	-------------------	-------------------	--------------	----------

Sprint-1	Dataset Collection	USN-1	Collect Dataset for building model.	9	High
Sprint-1	Image Pre-processing	USN-2	Perform Pre-processing techniques on the dataset.	8	Medium
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model.	10	High
Sprint-2		USN-4	Training the image classification model using CNN.	7	Medium
Sprint-3	Training and Testing the Model	USN-5	Training the model and testing the model's performance.	9	High
Sprint-4	Application Development	USN-6	Converting the input gesture image into English Alphabets.	8	Medium

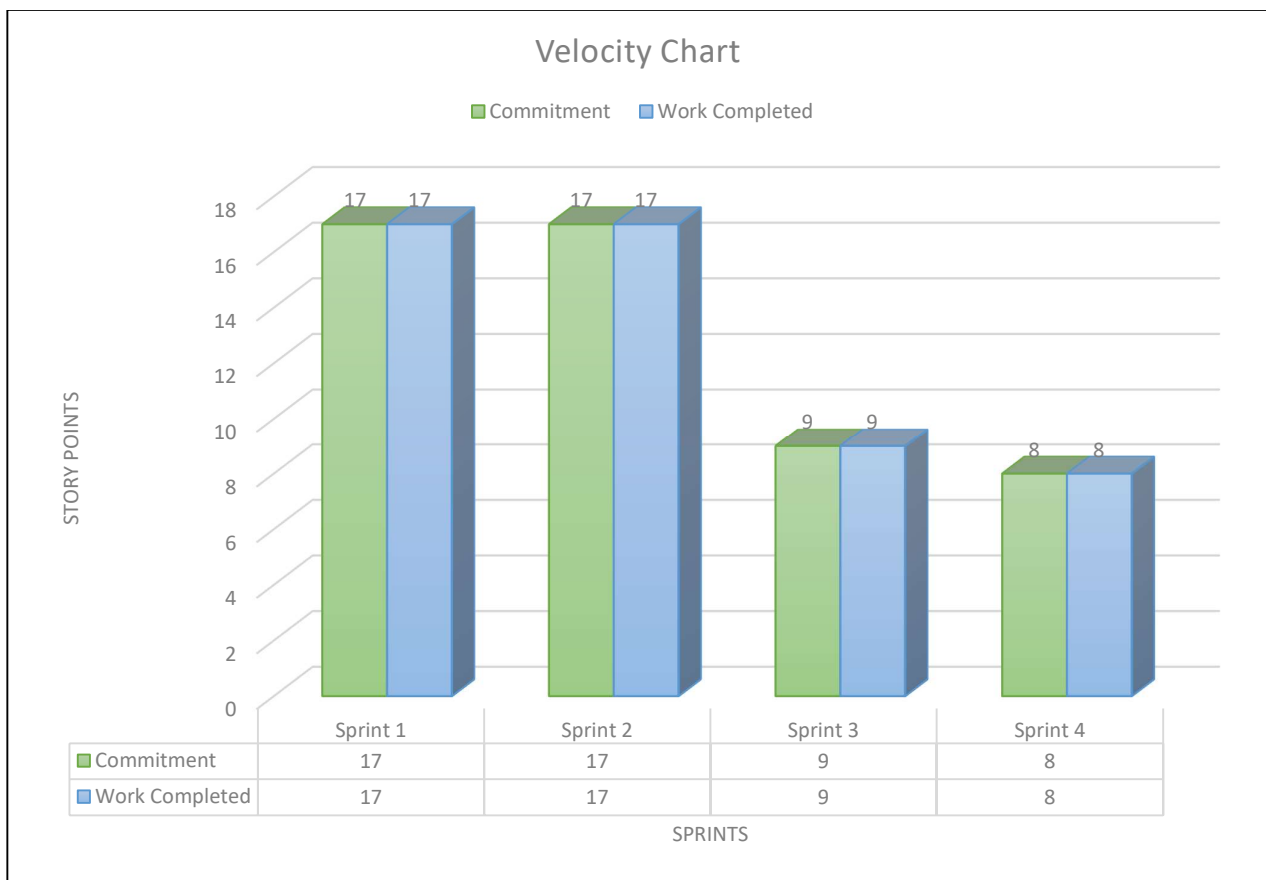
6.2. SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	17	6 Days	24 Oct 2022	29 Oct 2022	17	29 Oct 2022
Sprint-2	17	6 Days	31 Oct 2022	05 Nov 2022	17	05 Nov 2022
Sprint-3	9	6 Days	07 Nov 2022	12 Nov 2022	9	12 Nov 2022
Sprint-4	8	6 Days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

- Burndown Chart



- Velocity Chart



6.3. REPORTS FROM JIRA

	OCT							NOV							NOV							NOV							NOV				
	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Sprints	PA Sprint 1							PA Sprint 2							PA Sprint 3							PA Sprint 4											
PA-10 Dataset Collection and Image Preprocessing																																	
PA-11 Model Building																																	
PA-12 Training and Testing the Model																																	
PA-13 Application Development																																	

7. CODING & SOLUTIONING

7.1. IMAGE PREPROCESSING

- Image pre-processing includes zooming, shearing, flipping to increase the robustness of the model after it is built. Keras package is used for pre-processing images.
- Importing ImageDataGenerator Library to create an instance for which include shearing, rescale, zooming, etc to make the model robust with different types of images.

```
In [1]: from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\tensorflow\compat\tensorflow_stub\dtypes.py:543: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint8 = np.dtype [("qint8", np.uint8, 1)]
c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\tensorflow\compat\tensorflow_stub\dtypes.py:543: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint16 = np.dtype [("qint16", np.int16, 1)]
c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\tensorflow\compat\tensorflow_stub\dtypes.py:544: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint16 = np.dtype [("qint16", np.int16, 1)]
c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\tensorflow\compat\tensorflow_stub\dtypes.py:545: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint32 = np.dtype [("qint32", np.int32, 1)]
c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\tensorflow\compat\tensorflow_stub\dtypes.py:550: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_resource = np.dtype [("resource", np.uint8, 1)]
```

- Applying ImageDataGenerator Functionality To Train And Test Set

```
In [2]: x_train = train_datagen.flow_from_directory('dataset/training_set', target_size=(64,64), batch_size=300, class_mode='categorical', color_mode='rgb')
x_test = test_datagen.flow_from_directory('dataset/test_set', target_size=(64,64), batch_size=300, class_mode='categorical', color_mode='rgb')
Found 15750 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

7.2. MODEL BUILDING

- Importing The Required Model Building Libraries

```
In [3]: from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
```

- Initializing The Model

```
In [4]: model = Sequential()
WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorflow_backend.py:74: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.
```

- Adding The Convolution Layer

```
In [5]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))

WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf.random_uniform is deprecated. Please use tf.random.uniform instead.
```

- Adding The Pooling Layer

```
In [6]: model.add(MaxPooling2D(pool_size=(2,2)))

WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorflow_backend.py:3976: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.
```

- Adding The Flatten Layer

```
In [7]: model.add(Flatten())
```

- Adding The Dense Layers

```
In [8]: model.add(Dense(units=512,activation='relu'))
model.add(Dense(units=9,activation='softmax'))
```

- Compiling The Model

```
In [9]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorflow_backend.py:3295: The name tf.log is deprecated. Please use tf.math.log instead.
```

- Fit And Saving the Model

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

model.save('signlanguage.h5')

WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorflow_backend.py:986: The name tf.assign_add is deprecated. Please use tf.compat.v1.assign_add instead.

Epoch 1/10
24/24 [=====] - 17s 720ms/step - loss: 1.3771 - acc: 0.6265 - val_loss: 0.5072 - val_acc: 0.8809
Epoch 2/10
24/24 [=====] - 15s 635ms/step - loss: 0.3532 - acc: 0.9039 - val_loss: 0.2881 - val_acc: 0.9302
Epoch 3/10
24/24 [=====] - 15s 633ms/step - loss: 0.1826 - acc: 0.9522 - val_loss: 0.1999 - val_acc: 0.9471
Epoch 4/10
24/24 [=====] - 15s 645ms/step - loss: 0.1130 - acc: 0.9707 - val_loss: 0.1902 - val_acc: 0.9676
Epoch 5/10
24/24 [=====] - 15s 630ms/step - loss: 0.0773 - acc: 0.9802 - val_loss: 0.1742 - val_acc: 0.9653
Epoch 6/10
24/24 [=====] - 15s 641ms/step - loss: 0.0534 - acc: 0.9872 - val_loss: 0.1953 - val_acc: 0.9733
Epoch 7/10
24/24 [=====] - 15s 633ms/step - loss: 0.0366 - acc: 0.9917 - val_loss: 0.1920 - val_acc: 0.9698
Epoch 8/10
24/24 [=====] - 15s 637ms/step - loss: 0.0314 - acc: 0.9936 - val_loss: 0.2024 - val_acc: 0.9716
Epoch 9/10
24/24 [=====] - 15s 630ms/step - loss: 0.0271 - acc: 0.9939 - val_loss: 0.2371 - val_acc: 0.9738
Epoch 10/10
24/24 [=====] - 15s 634ms/step - loss: 0.0192 - acc: 0.9965 - val_loss: 0.1944 - val_acc: 0.9747
```

7.3. TESTING THE MODEL

- Importing The Packages and Loading the Saved Model


```

In [1]: from keras.models import load_model
        from keras.models import Sequential
        import numpy as np
        import cv2

        # Load the model
        model = load_model('signlanguage.h5')

        # Load the test image
        img = cv2.imread('dataset/test_set/G/1.png')

        # Pre-process the image
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = img / 255.0

        # Predict the class
        prediction = model.predict(img)
        print(prediction)
        predictions = model.predict_classes(img)
        print(predictions)

In [2]: model = load_model('signlanguage.h5')

end.py:3976: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.
WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorflow_backend.py:174: The name tf.get_default_session is deprecated. Please use tf.compat.v1.get_default_session instead.
WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorflow_backend.py:181: The name tf.ConfigProto is deprecated. Please use tf.compat.v1.ConfigProto instead.
WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorflow_backend.py:186: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.
WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
WARNING:tensorflow:From c:\users\adars\appdata\local\programs\python\python37\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: adadelta_support_wrapper.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where

```

- Loading the Test Image, Pre-Processing it And Prediction

```

In [19]: from skimage.transform import resize

        def detect(frame):
            img = resize(frame, (64, 64, 1))
            img = np.expand_dims(img, axis=0)
            if np.max(img) > 1:
                img = img / 255.0
            prediction = model.predict(img)
            print(prediction)
            predictions = model.predict_classes(img)
            print(predictions)

In [21]: frame = cv2.imread(r"dataset/test_set/G/1.png")
        data = detect(frame)

[[1.1529493e-09 1.6801257e-12 3.0758306e-07 3.6168924e-08 2.1814937e-11
  6.9361130e-09 9.9995184e-01 4.7746969e-05 3.6307211e-09]]
[6]

```

- The output [6] in the above image represents the index value in the array ['A','B','C','D','E','F','G','H','I'].
- Thus, the predicted alphabet is G.

7.4. FLASK APPLICATION

- Loading the required packages

```

import numpy as np
import cv2
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.backend import set_session
from flask import Flask, render_template, Response
import tensorflow as tf
from gtts import gTTS
global graph
global writer
from skimage.transform import resize

```

- Initializing graph, loading the model, initializing the flask app and loading the video.
- Graph element is required to work with TensorFlow. So, graph element is created explicitly.

```
graph = tf.get_default_graph()
model = load_model('signlanguage1.h5')
vals = ['A','B','C','D','E','F','G','H','I']
app = Flask(__name__)
print("[INFO] accessing video stream...")
camera = cv2.VideoCapture(1)
camera.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
camera.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)
pred=""
```

- Configuring the home page

```
@app.route('/')
def index():
    return render_template('index.html')
```

- Pre-processing the frames captured from the camera

```
def detect(frame):
    global pred
    global graph
    img = resize(frame, (64,64,1))
    x = image.img_to_array(img)
    x = np.expand_dims(x,axis=0)
    with graph.as_default():
        predictions = model.predict_classes(x)
    print(predictions)
    pred=vals[predictions[0]]
    print(pred)
```

- Video Feed call from the HTML PAGE

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

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

7.5. HTML PAGE

- HTML page to display the processed video on the screen, so that the person can show signs which can be detected.


```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
  <title>VHearuU</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.12.0/css/all.css">
  <link rel="stylesheet" href="static/css/banner-heading-image.css">
  <link rel="stylesheet" href="static/css/Navbar-Centered-Brand.css">
  <link rel="stylesheet" href="static/css/styles.css">
</head>
<body>
<nav class="navbar navbar-light navbar-expand-md py-3" style="background: #429691;">
  <div class="container">
    <div></div><div><div class="navbar-brand d-flex align-items-center" href="#"><div style="color: #ffffff; font-style: oblique; text-align: center;font-family: Arial Black"><strong> Real-Time Communication
      System Powered By AI&nbsp;For Specially Abled - Team ID: PNT2022TMID09233</strong></div></div>
    </div></div>
  </nav>

<section>
  <div class="container">
    <div class="accordion text-white" role="tablist" id="accordion-1">
      <div class="accordion-item" style="font-style: oblique; background: #429691;">
        <h2 class="accordion-header" role="tab"><button class="accordion-button collapsed"
          data-bs-toggle="collapse" data-bs-target="#accordion-1 .item-2" aria-expanded="false"
          aria-controls="accordion-1 .item-2"
          style="font-style: oblique; background: #cc7931;color: #ffffff;"><strong>Project: Real-Time Communication System Powered by AI for Specially Abled</strong></h2>
        <div class="accordion-collapse collapse item-2" role="tabpanel" data-bs-parent="#accordion-1">
          <div class="accordion-body">
            <p class="mb-0">Team ID: PNT2022TMID09233<br><br>VHearu Developed By,<br><br>1. <strong>Abubucker Siddique</strong> 310619104001<br>2.
              <strong>Adarsh Srinivas</strong> 310619104003<br>3. <strong>Ashwath</strong> 310619104012<br>4. <strong>Jayasurya</strong> 310619104044
            </p>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
<div class="modal fade" role="dialog" tabindex="-1" id="modal-1">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h4 class="modal-title">American Sign Language - Alphabets</h4><button type="button"
          class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body"></div>
      <div class="modal-footer"><button class="btn btn-secondary" type="button"
        data-bs-dismiss="modal">Close</button></div>
    </div>
  </div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

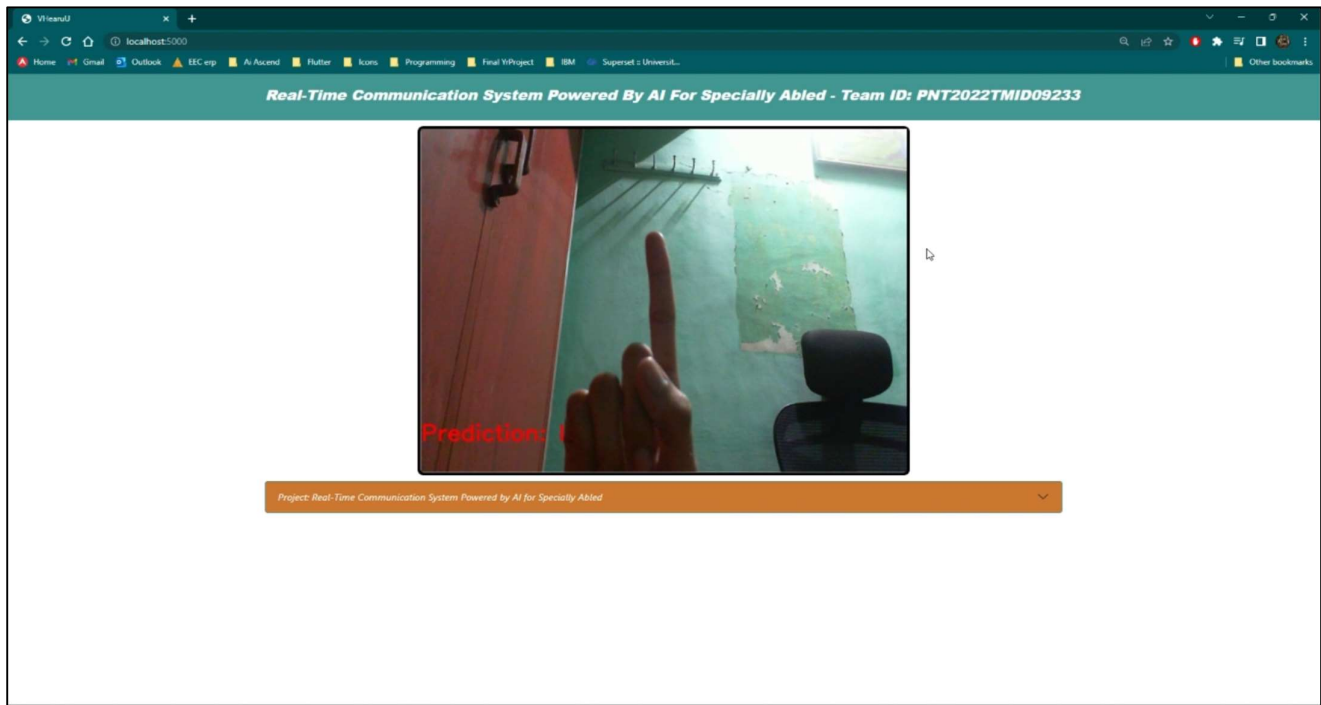
8. TESTING

8.1. TEST CASES

Test Case ID	Test Scenario	Steps to Execute	Expected Result	Actual Result
1	Verify if user is able to provide camera access.	1. Enter URL and click go. 2. Give Camera Access.	Camera is On.	Working as expected.
2	Verify if user is able to get the desirable prediction for the gesture.	1. Enter URL and click go. 2. Give Camera Access. 3. Make Gesture in front of camera.	Alphabet is predicted for the gesture.	Working as expected.

8.2. USER ACCEPTANCE TESTING

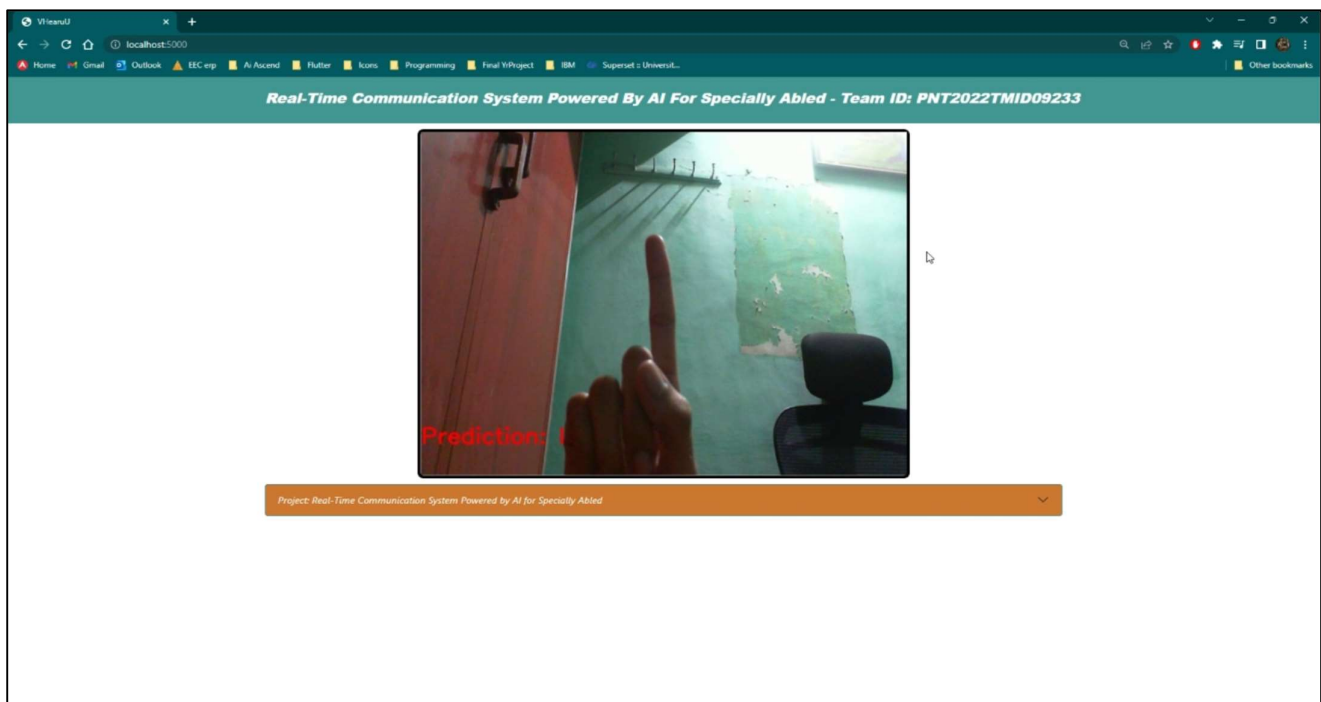
- The project developed was tested by an end user and the application converts the gestures to its respective alphabet accurately.



9. RESULTS

9.1. PERFORMANCE METRICS

- The project is developed using CNN (Convolutional Neural Network) model.
- The said model is trained using the dataset consisting of 15750 images of ASL (American Sign Language) alphabets.
- The said model is then test using the dataset consisting of 2250 images of ASL (American Sign Language) alphabets.
- The dataset of ASL (American Sign Language) alphabets range from 'A' to 'I'.
- During Real-Time, the gestures are fed to the model and the respective alphabet is predicted.



10. ADVANTAGES & DISADVANTAGES

- Advantages
 - The application is conveniently simple for the end user.
 - The user interface is not complex.
- Disadvantages
 - The dataset is limited.
 - The alphabets only range from 'A' to 'T'.
 - As of now, only static gestures are converted.

11. CONCLUSION

The main objective of this project is to develop gesture recognition so that the deaf can communicate with normal individuals. One of the crucial tasks is the extraction of features, and various gestures should yield various, effectively distinguishable characteristics. To identify the character from the gesture images, we used a trained dataset for the CNN algorithm. These features combined with a labelled data enable accurate real-time ASL alphabet recognition. Our analysis found that accuracy is influenced by a variety of elements, including the camera, dataset, and approach. The accuracy drastically declines in low light and noisy backgrounds.

12. FUTURE SCOPE

The proposed system can be translated into multiple languages, enhancing its dependability and effectiveness. In the near future, it might only be accessible through mobile devices, making the system more convenient and portable. This system is unable to detect gestures made with both hands. Therefore, detecting gestures done with both hands could be a future project.

13. APPENDIX

13.1. SOURCE CODE

- Source Code is available in the GitHub link provided in Section 13.2.

13.2. GITHUB & PROJECT DEMO LINK

- GitHub: <https://github.com/IBM-EPBL/IBM-Project-35135-1660281810>
- Project Demo Link: <https://www.loom.com/share/0bd6f8a58b2c405896a9ebd7b70915d7>