REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

A PROJECT REPORT

Submitted by

HELAN. J **(Reg. No 912619104015)**

NIROSHIKA. R **(Reg. No 912619104014)**

RANJANI. K **(Reg. No 912619104019)**

ROOPINA. R **(Reg. No 912619104021)**

In partial full filment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

SRI BHARATHI ENGINEERING COLLEGE FOR WOMEN

KAIKKURICHI, PUDUKKOTTAI-622 303

ANNA UNIVERSITY: CHENNAI600 025

DEC 2022

PROJECT REPORT FORMAT

1. INTRODUCTION

1.1. Project Overview

1.2. Purpose

2. LITERATURE SURVEY

2.1. Existing problem

2.2. References

2.3. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas

3.2. Ideation & Brainstorming

3.3. Proposed Solution

3.4. Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1. Functional requirement

4.2. Non-Functional requirements

5. PROJECT DESIGN

5.1. Data Flow Diagrams

5.2. Solution & Technical Architecture

5.3. User Stories

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

1. INTRODUCTION

1.1. Project Overview

People get to know one another by sharing their ideas, thoughts, and experiences with those around them. There are numerous ways to accomplish this, the best of which is the gift of "Speech." Everyone can very convincingly transfer their thoughts and understand each other through speech. It will be unjust if we overlook those who are denied this priceless gift: the deaf and dumb. In such cases, the human hand has remained the preferred method of communication.

1.2. Purpose

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.he project aims to develop a system that converts the sign language into a human hearing voice in the desired language or text to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making using 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. LITRATURE SURVEY

S.No	Journal and Authors	Title	Methodology	Pros	Cons
1.	International Journal of Advanced Trends in Computer Science and Engineering **Authors** - Bayan Mohammed Saleh**,** -Reem Ibrahim Al Beshr's -Muhammad UsmanTariq	D-Talk: Sign Language Recognition System for People with Disability using Machine Learning and Image Processing - 2020	-Multilayer Network -Neural network	This paper focuses on the classification methods used in prior recognition scheme for sign Recognition.	The system is very sensitive. It catches any element in the box. So, the user must be careful to have a blank background
2.	International Journal of Innovative Research in Science, Engineering and Technology **Authors** -Kedar Potdar -Gauri Nagavkar	Real-time Communication System for the Deaf and Dumb – 2017	-Flex sensors -Arduino	These electrical signals are then processed using an Arduino micro controller and a Python-based backend for text-to-speech conversion	TTS functionality can be eliminated by adding a portable computer like the Raspberry Pi

3.	American-Eurasian Network for Scientific Information **Authors** -D.Narashiman -T.Mala	Sign gesture representation using and-or tree -2015	-Sign gesture rendering -Dynamic sign generation	The advantage of the proposed method over the earlier method is that the location of the hand position is clearly mentioned.	unavailability of good sign language tutor, lack of importance to that community
4.	Intelligent Technologies for Communication, Learning and Teaching	Assistive & augmentive Communication for the Disabled -2014	-Viola -Jones Face	An enhanced interpersonal-human interaction for people with special needs,	This would be another domain for further research on how a machine

2.1 Existing Problem

Some 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 has low 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. References

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

[2] Manduchi, R., Kurniawan, S., & Bagherinia, H. (2010, October). Blind guidance using mobile computer vision: A usability study. In Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility (pp. 241-242).

[3] Ivanchenko, V., Coughlan, J., Gerrey, W., & Shen, H. (2008, October). Computer vision-based clear path guidance for blind wheelchair users. In Proceedings of the 10th international ACM SIGACCESS onference on Computers and accessibility (pp. 291-292).

[4] Johnson, A., Grønli, T. M., & Bygstad, B. (2012). Making touch-based mobile phones accessible for the visually impaired. Norsk

informatikkonferanse,(Bodø, Norway, 2012).

[5] Matusiak, K., Skulimowski, P., & Strurniłło, P. (2013, June). Object recognition in a mobile phone application for visually impaired users. In 2013 6th International Conference on Human System Interactions (HSI) (pp. 479-484). IEEE.

[6] Hermus, K., & Wambacq, P. (2006). A review of signal subspace speech enhancement and its application to noise robust speech recognition. EURASIP Journal on Advances in Signal Processing, 2007(1), 045821.

[7] Dimitrov, V., Jullien, G., & Muscedere, R. (2017). Multiple-base number system: theory and applications.CRC press.

[8] Huyan, Z., Xu, L., Fang, S., Liu, Z., Zhang, X., & Li, L. (2014). Field information acquisition system research based on offline speech recognition. Int. J. Database Theory Application, 7, 45-58.

[9] Oman khanlen, A. E., & Ogaga-Oghene, J. (2013). The Dynamics of Global Strategy and Strategic Alliances in International Trade and Investment. INTERNATIONAL JOURNAL OF RESEARCH IN

COMPUTER APPLICATION & MANAGEMENT, 3(12), 41-48.

[10] Kamble, K., & Kagalkar, R. (2014). A review: translation of text to speech conversion for Hindi language. International Journal of Science and Research (IJSR) Volume, 3.

[11] Arora, S. J., & Singh, R. P. (2012). Automatic speech recognition: a review. International Journal of Computer Applications, 60(9).

[12] BELGHIT, H., & BELLARBI, A. Object Recognition Based on ORB Descriptor for Marker less

Augmented Reality.

[13] Gill, J. (2000). Personal electronic mobility devices. Information for Professionals Working with Visually Disabled People. http://www. tiresias. org.

[14] Chen, C., & Raman, T. V. (2009). Announcing eyes-free shell for Android. Retrieved December, 21, 2016.

[15] Coughlan, J., & Manduchi, R. (2009). Functional assessment of a camera phone-based way finding system operated by blind and visually impaired users. International Journal on Artificial IntelligenceTools, 18(03), 379-397.

[16] Coughlan, J., & Manduchi, R. (2007). Color targets: Fiducials to help visually impaired people find their way by camera phone. EURASIP Journal on Image and Video Processing, 2007, 1-13.

[17] Kumar, A., & Chourasia, A. (2018). Blind Navigation System Using Artificial Intelligence. International Research Journal of Engineering and Technology, 5(3).

[18] Jiang, R., Lin, Q., & Qu, S. (2016). Let Blind People See: Real-Time Visual Recognition with Results Converted to 3D Audio. Report No. 218, Standard University, Stanford, USA.

[19] Bigham, J. P., Jayant, C., Miller, A., White, B., & Yeh, T. (2010, June). VizWiz:: LocateIt-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.

[20] Bigham, J. P., Jayant, C., Miller, A., White, B., & Yeh, T. (2010, June). VizWiz:: LocateIt-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.

[21] Manduchi, R., Kurniawan, S., & Bagherinia, H. (2010, October). Blind guidance using mobile computer vision: A usability study. In Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility - (pp. 241-242).

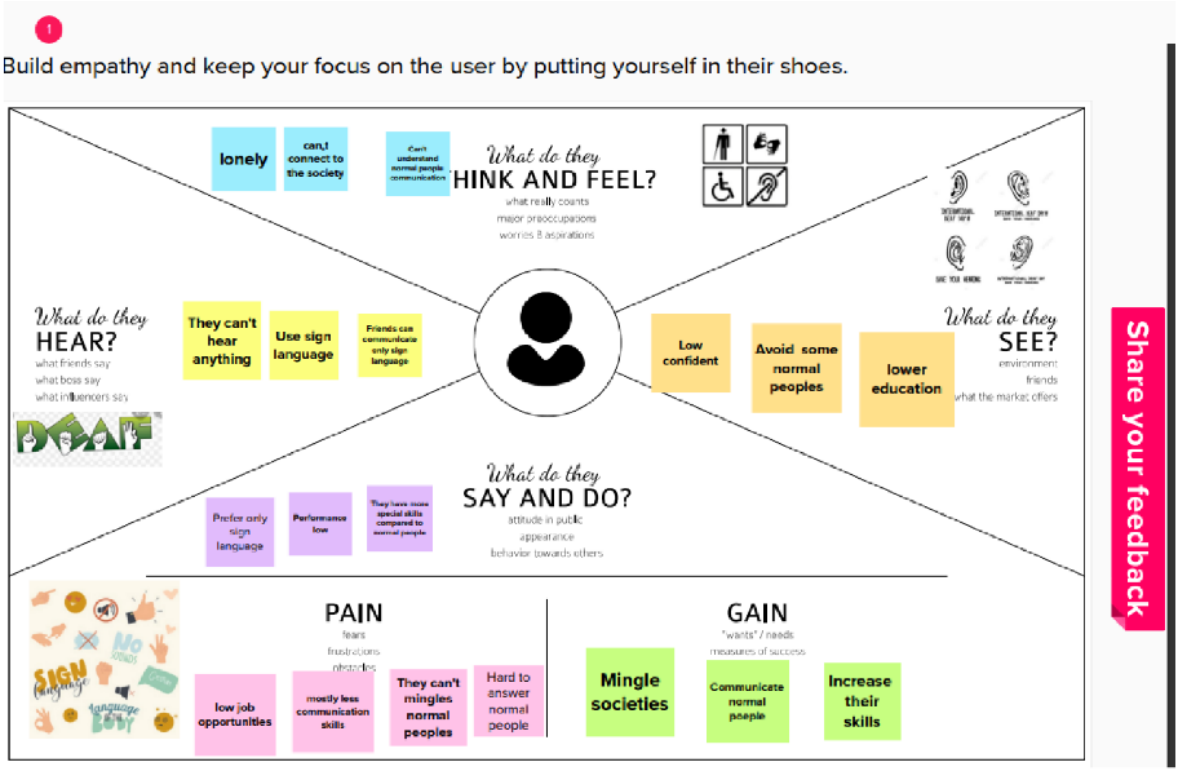
2.3. Problem Statement Definition

The main problem statement is to acute design which can provide voice to voiceless indicate glove. It shows that with the help or use of these smart glove there won't be any miscommunication between two distinct groups and they will have the scope to discuss fluently. The glove is developed with flex sensors, contact sensors and accelerometer to estimate flexion of the fingers and the turnof hand. With different signals,hand motion assumes an essential part, as it communicates the client's perspective in less time. The aim is to only play out these mind blogging calculation and activities on the server and create the discourse on cell phones.It sometime uncomfortable to the users

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community



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

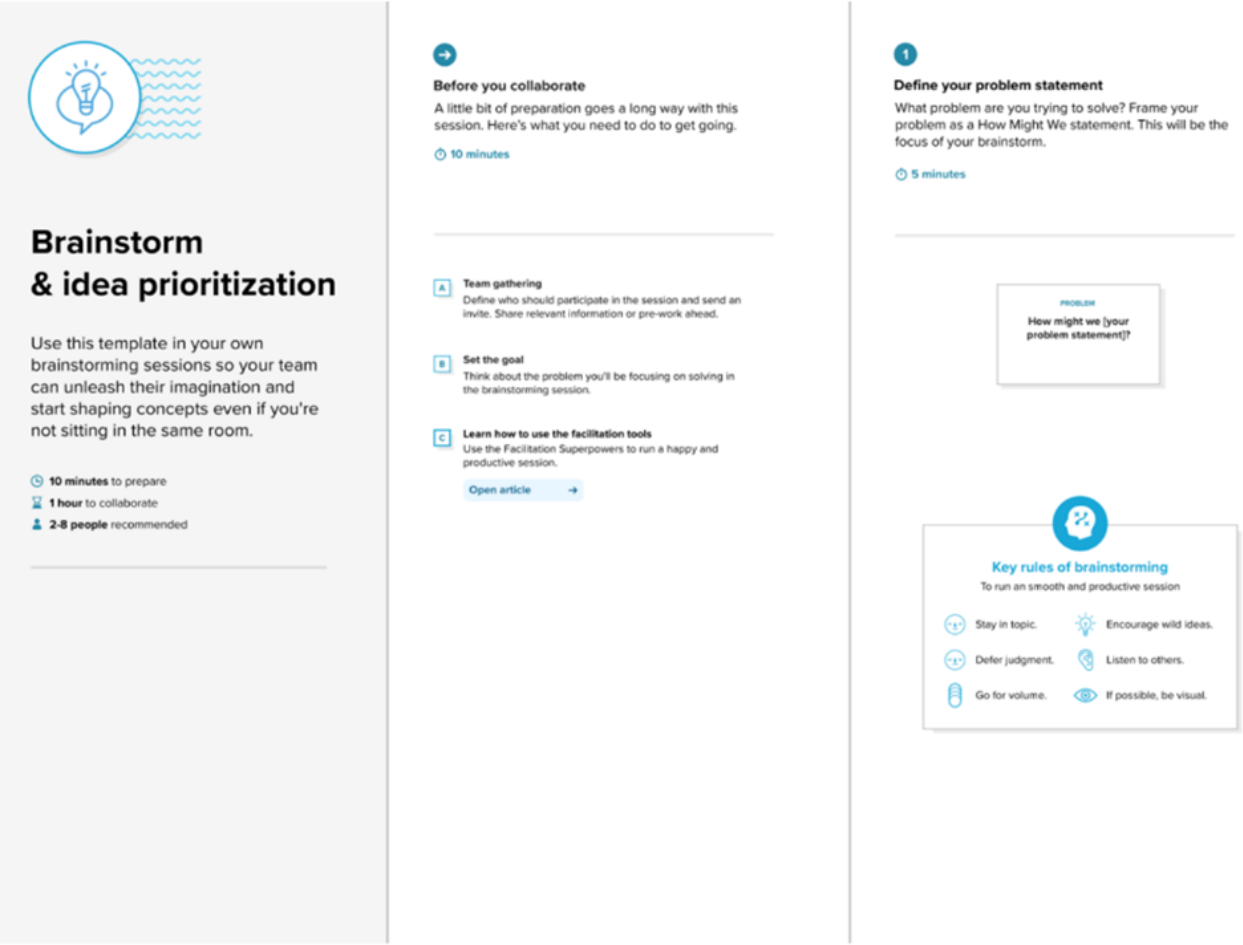
3.2. IDEATION & BRAINSTORM

Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge

.

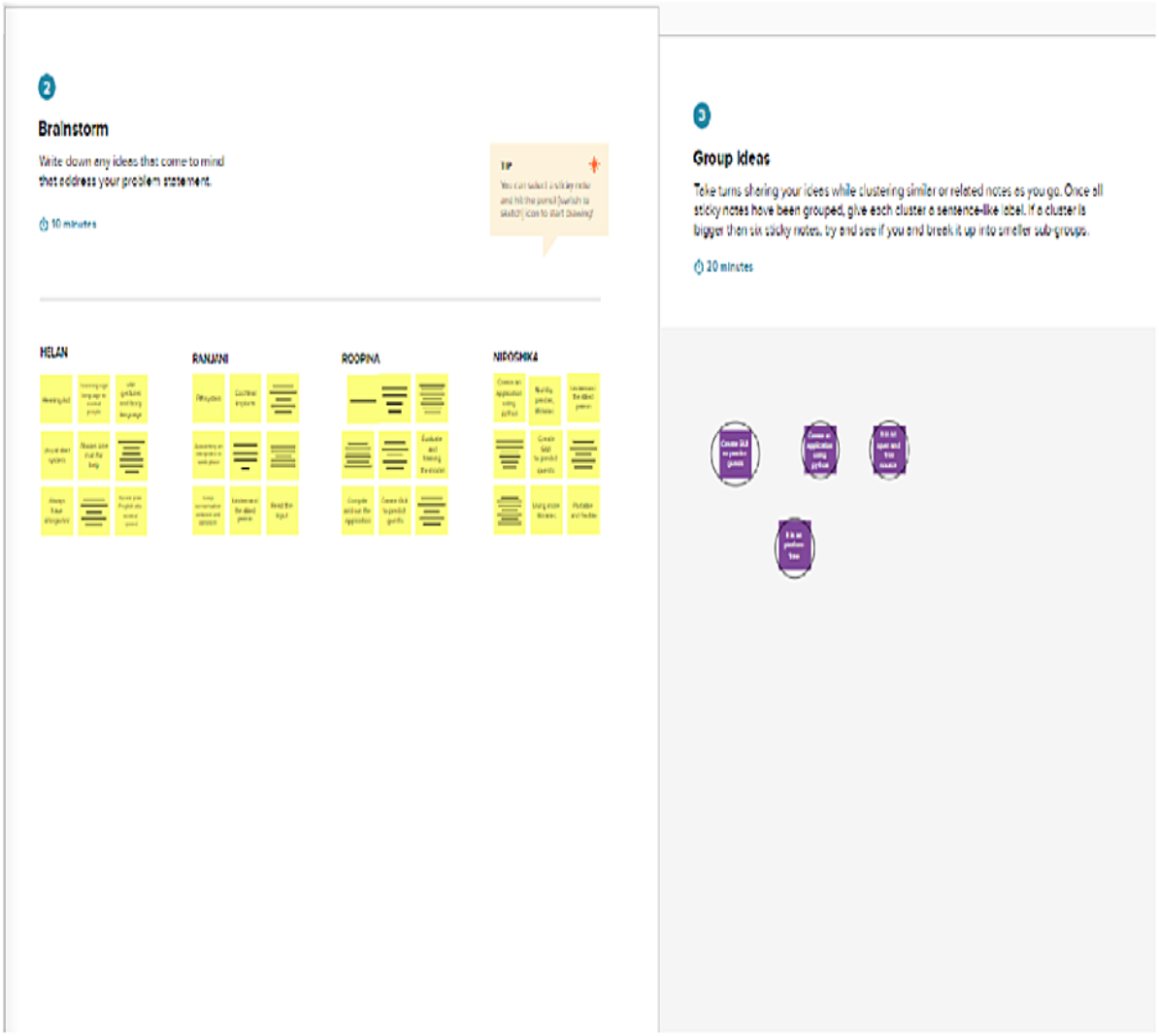
Step-1: Team Gathering,Collaboration and select the problem statement

Step-1 is define Team Gathering,Collaboration and select the problem statement concept of the project.



Step2: Brainstorm Idea, Listening and Grouping

Step 2 is define Brainstorm idea, listening and grouping concept of the project.



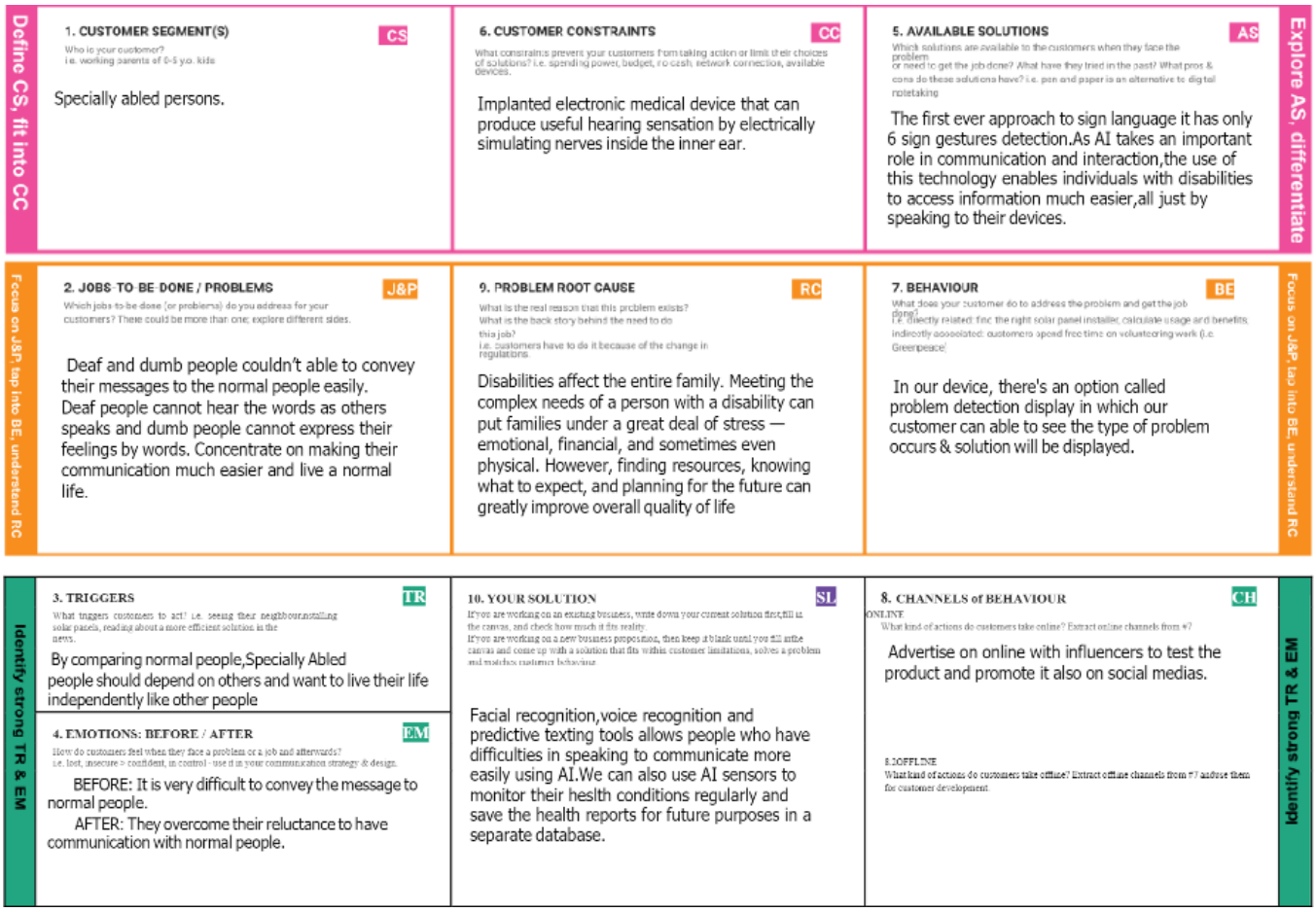
3.3. Proposed Solution

The following table is describe the details of proposed system.

S. No.	Parameter	Description
1.	Problem Statement(Problem to be solved)	Sign Language is a visual means of communicating using gestures, facial expressions, and body language with specially- Abled. Since normal people are not trained in sign language, in times of emergency conveyingtheir message is very difficult. Hence, there is a need for a system that recognizes different signs and empowersthem in communicating with normal people
2.	Idea / Solution description	The idea is to create an end-end application that predicts the ISL signs from a live video andtranslates the same to voice such that conversing is at ease
3.	Social Impact/ Customer Satisfaction	● Communication is achieved without the helpof additional humanintervention. ● No additional hardware support is needed touse the application ● Improve their career opportunities in the industry can provide instantresults to users
4.	Novelty / Uniqueness	We use a stopping symbol to group the lettersinto a word and generate a sentence and the resulted prediction is converted into speech to convey the information in a convenient manner

3.4. Problem Solution fit

The following table is describe the details of problem solution of this project**.**



4. REQUIREMENT ANALYSIS

4..1 Functional Requirements

The Functional Diagram Dsfines the details of how man requirements are te developer needed.

FR. NO	FUNCTIAL REQUIREMENTS	SUB REQUIREMETNS
FR-1	User Registration	Registration Through Form Registration Through Mail
FR-2	User Confirmation	Confirmation Via Email Confirmtion Via OTP
FR-3	User Login	Login Through Form Login Through Mail
FR-4	Image Upload	Image Upload Via Drog & Drop Image Upload Via Checking From Local SubSystem
FR-5	Text Entry	Text Entry By Typing Text Entry By Copy, Paste
FR-6	Conversion	Sign-Language to Speech Conversion Text-to Sign Language Conversio
FR-7	Activity Log	View List OF Active Users View Log Of Translation Performed

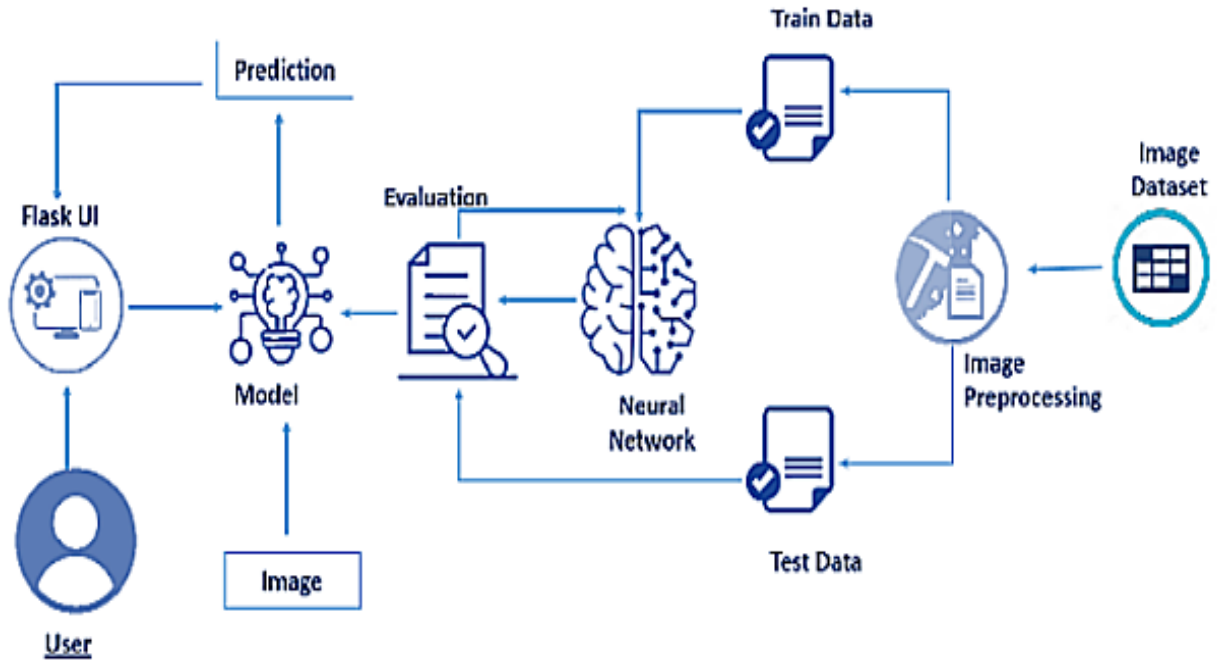
4.2. Non-Functional requirements

FR. NO	FUNCTIAL REQUIREMENTS	DESCRIPTION
NFR-1	Usability	The application can be used by clients without priority knowledge of machine learning algorithm.
NFR-2	Security	Access is provided only to register users, either through a username-password
FNR-3	Reliability	The application has an extremely low failure rate /down time and has a clear, predictable flow of operation
NFR-4	Performance	The application provides accurate translation from sign-language in to speech as well as text-to -sign language
NFR-5	Availability	Clients can use application to perform conversation provide they have a stable internet connection
NFR-6	Scalability	The application seamlessly and efficient handles a growing number of clients/ users without any significant reduction in performance

5. PROJECT DESIGN

5.1. Data Flow Diagrams

A Data flow diagram(DFD) is a traditional visual representation of the information flows with in a system. A neat and clear DFT 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.



5.2 Solution & Technical Architecture

Technical Architecture: The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2 Real-Time Communication System Powered by AI for Specially Abled

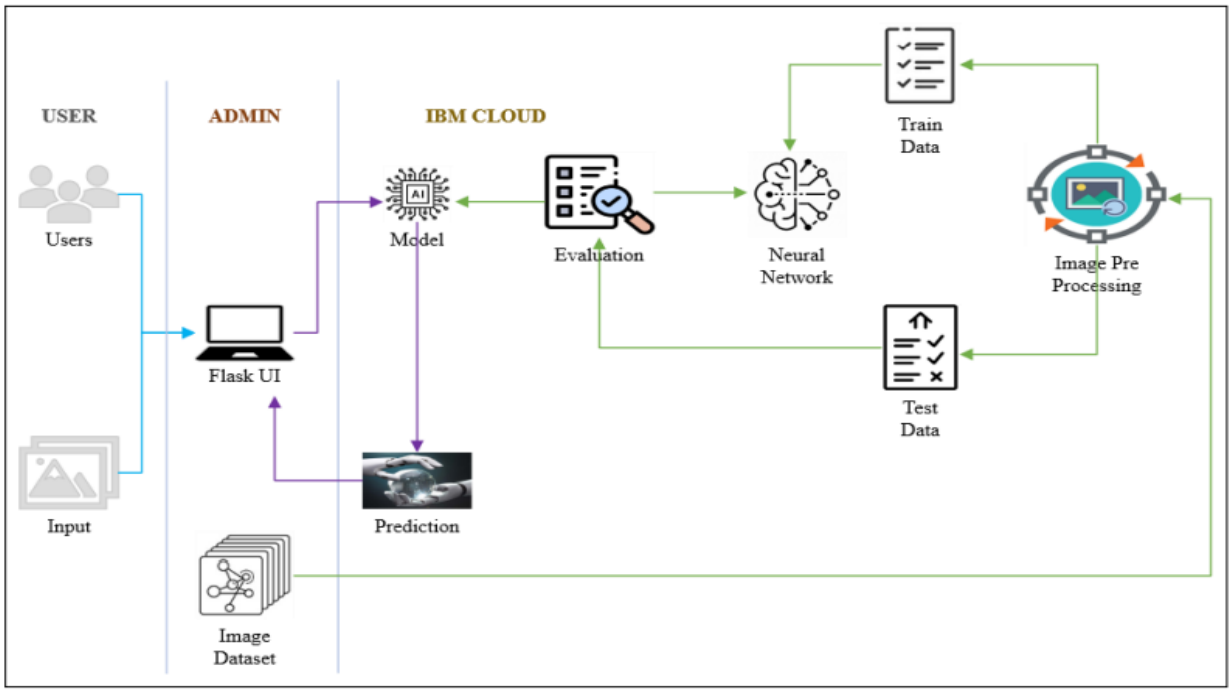


Table-1 : Components & Technologies:

S. 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 users that use your site and gather information.	Using the cloud, it can be executed.
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.

5.3 User Stories

User Type	Functional Requirement (Epic)	User story number	User story/task	Acceptance criteria	priority	Release
Customer (deaf people)	Registration	USN-1	As a user, I can register for the application by entering my email,password ,and confirming my password	I can registered my account	High	Sprint 1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can received conformation gmail and click confirm	High	Sprint 2
	Login	USN-3	As a user, I can log into the application through gmail	I can registered and access the dashboard with gmail login	Low	Sprint 1
		USN-4	As a user, I can see my application and made changes in any browsers	I can login and see my account at anywhere	Medium	Sprint 2
	Dashboard	USN-5	As a user, I can create my account in a given dashboard	I can access my account/dash board	High	Sprint 1

Customer (dumb people)	Registrati on	USN-6	As a user, I can register my application through gmail	I can registered my account	High	Sprint 2
		USN-7	As a user, I can receive confirmation mail &get verification code from OTP & gmail	I can receive confirmation mail and click confirm	Low	Sprint 2
	Login	USN-8	As a user, I can log into my account by any web browsers	I can login and see my account	Medium	Sprint 1
	Dashboard	USN-9	As a user, I can create my account in a given dashboard	I can create my account & access into dashboard	High	Sprint 2

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

Milestone	**Functional Requirement (Epic)**	**Milestone Story Number**	**Milestone Story / Task**
Milestone 1	Data Collection	M1	We're collecting dataset for building our project and creating two folders, one for training and another one for testing.
Milestone 2	Image Preprocessing	M2	Importing image data generator libraries and applying image data generator functionality to train the test set
Milestone 3	Building Model	M3	Importing the model building libraries, Initializing the model, Adding Convolution layers, Adding the Pooling layers, Adding the Flatten layers, Adding Dense layers, Compiling the model Fit and Save the model.
Milestone 4	Testing Mode	M4	Import the packages first. Then we save the model and Load the test image process 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 Model
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

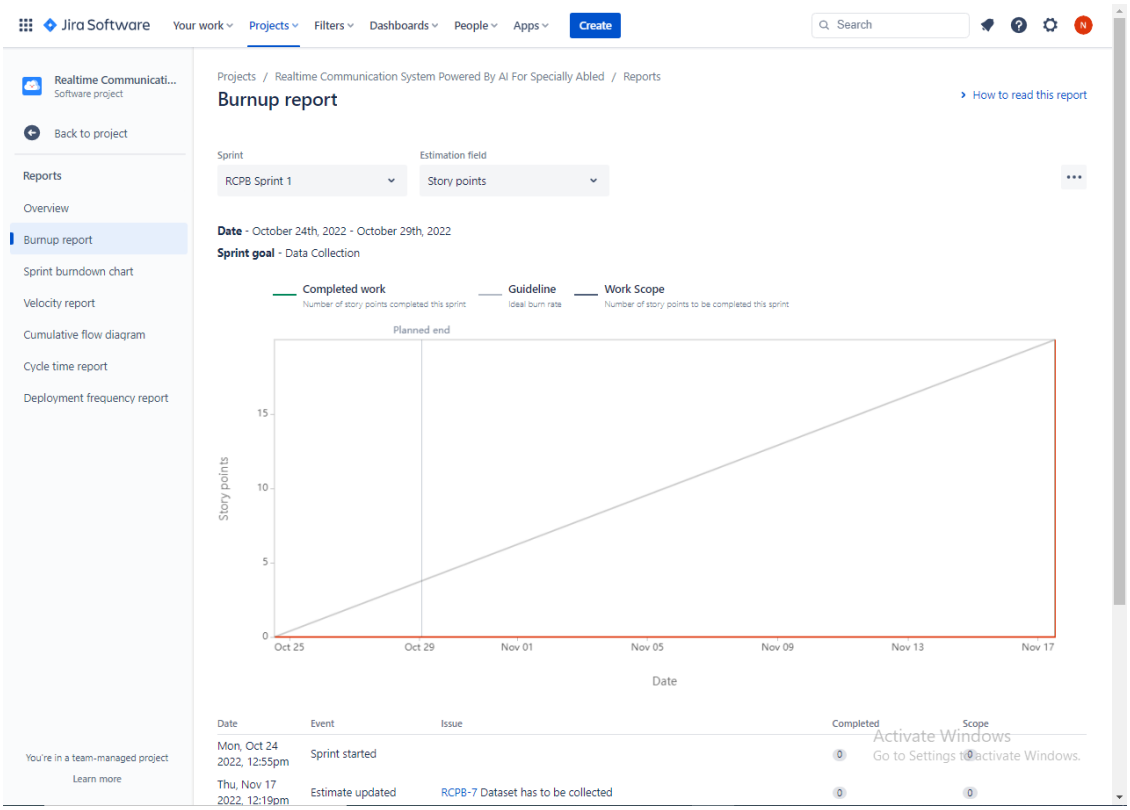
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint 1	Data Collection	USN-1	Dataset has to be collected	12	High	HELAN NIROSHIKA
Sprint 1	Image Preprocessing	USN-2	Collected images has to be preprocessed	8	Medium	HELAN RANJANI ROOPINA
Sprint 2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	12	High	RANJANI ROOPINA
Sprint 2	Model Building	USN-4	Training the image classification	8	Medium	HELAN NIROSHIK

			model usingCNN			RANJANI
Sprint 3	Training & Testing	USN-5	Training the model and testing the model's performance	20	High	ROOPINA RANJANI NIROSHIKA
Sprint 4	Implementation of the application	USN-6	Converting the input sign language images into English	20	High	HELAN

6.3. Reports from JIRA

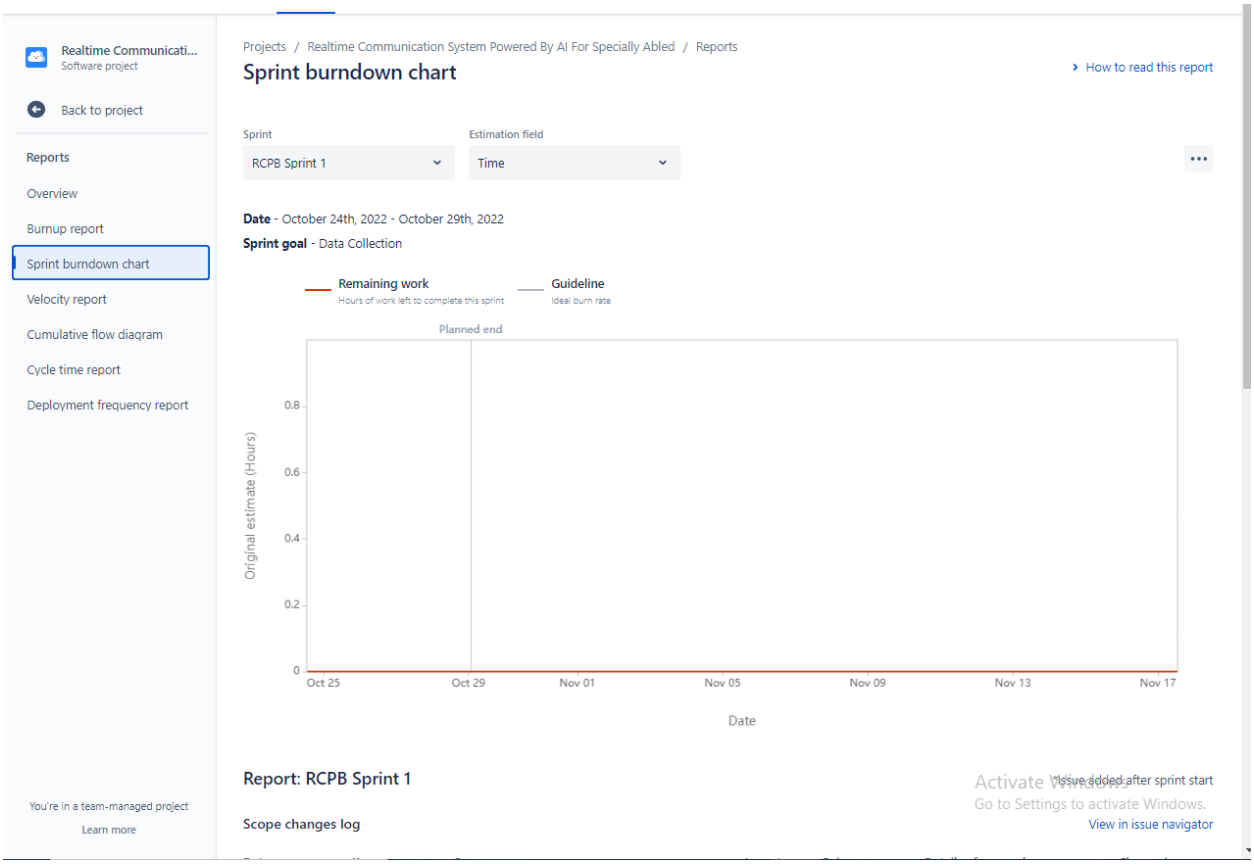
Burn up Report

The JIRA Reports tells the details of burn-up report by using all the sprint activities.



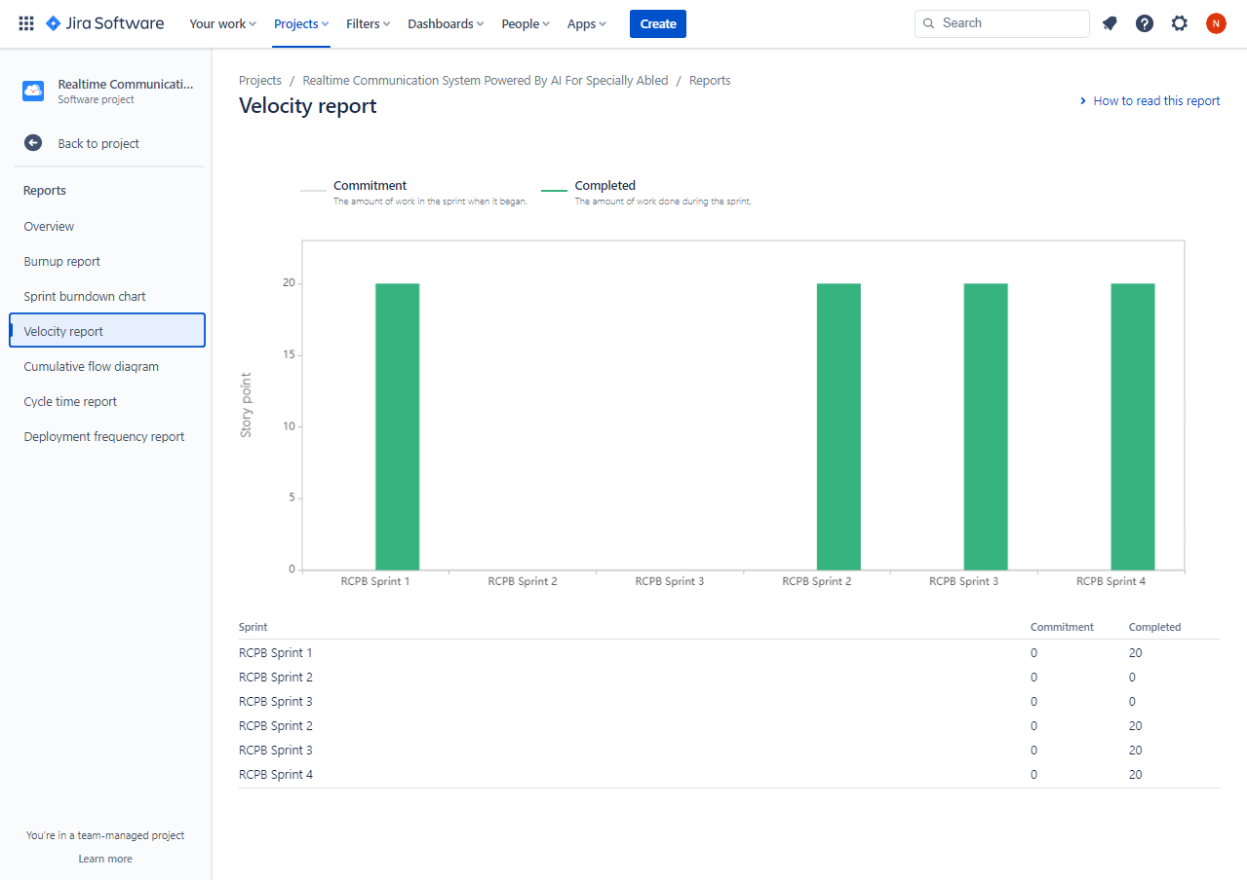
Sprint Burndown Chart

The JIRA file tells the details of sprint burndown by using all the sprints



Velocity Report

The JIRA file tells the details of how the velocity is working by the use of all sprint



Sprint	Commitment	Completed
RCPB Sprint 1	0	20
RCPB Sprint 2	0	0
RCPB Sprint 3	0	0
RCPB Sprint 2	0	20
RCPB Sprint 3	0	20
RCPB Sprint 4	0	20

7. CODING & SOLUTIONING

7.1. Feature 1

7.1.1. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It is high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program molecularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

7.1.2. CNN

In deep learning, a convolution neural network (CNN, or ConvNet) is a class of artificial neural network (ANN), most commonly applied to analyze visual imagery. CNNs are also known as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivalent responses known as feature maps.

Counter-intuitively, most convolutional neural networks are not invariant to translation, due to the downsampling operation they apply to the input. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language

processing, brain–computer interfaces, and financial time series.

CNNs are regularized versions of multi layer perceptrons. Multi layer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to over fitting data.

7.1.3 Python Flask

Flask is a web framework, it’s a Python module that lets you develop web applications easily. It’s has a small and easy-to-extend core: it’s a micro framework that doesn’t include an ORM (Object Relational Manager) or such features.

It does have many cool features like url routing, template engine. It is a WSGI web app framework.

To jumpstart a coding and programming career, it is crucial to master core concepts like a programming language and since it’s all about Python programming, it is paramount that developers understand and have great expertise in the basic fundamentals of Python programming. Various concepts like data structure, exception handling, object-oriented programming, and more are important to excel in a programming career.

7.2. Feature 2

7.2.1. IBM Cloud

IBM Cloud is a suite of cloud computing services from IBM that offers both platform as a service (PaaS) and infrastructure as a service (IaaS).

IBM Cloud Paas, which is based on the open source cloud platform Cloud Foundry

-- developers can use IBM services to create, manage, run and deploy various types of applications for the public cloud, as well as for local or on-premises environments. IBM Cloud supports various programming languages, such as Java, Node.js, PHP and Python and extends to support other languages.

7.2.2. IBM Watson

The Watson is designed to make it easy for you to use data from diverse sources, trust the recommendations and predictions from your AI models, and get more value from your AI, faster. With Watson, you have access to the most complete portfolio of AI capabilities for business, whether it’s tools for building your own models, pre-built applications to accelerate time to value, or access to a robust ecosystem of partners across multiple industries.

8. TESTING

8.1. Test Cases

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on “HOW” to validate a particular test objective / target, which when followed will tell us if the expected behaviour of the system

S. NO	Scenario	Input	Exceed output	Actual Output
1.	Dataset	train and test image	Image can be predicted by compared to dataset.	Dataset are stored in the database
2.	Image processing	sign images	Text letters are come in the output	Sign image converted in to text letters

8.2. User Acceptance Testing

This sort of testing is carried out by users, clients, or other authorized bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or program. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT).

9. RESULTS

9.1 Performance Metrics

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 recognize the equivalent Alphabet is shown on the screen.

10. ADVANTAGES & DISADVANTAGES

Advantages:

1. It is possible to create a mobile application to bridge the communication gap between deaf and dumb persons and the general public.

2. As different sign language standards exist, their dataset can be added, and the user can choose which sign language to read.

Disadvantages:

1. The current model only works from alphabets A to I.

2. In absence of gesture recognition, alphabets from J cannot be identified as they require some gesture input from the user.

3. 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.

11. CONCLUSION

Sign language is a useful tool for facilitating communication between deaf and hearing people. Because it allows for two-way communication, the system aims to bridge the communication gap between deaf people and the rest of society. The proposed methodology translates language into English alphabets that are understandable to humans.

This system sends hand gestures to the model, who recognizes them and displays the equivalent Alphabet on the screen. Deaf people can use their hands to perform sign language, which will then be converted into alphabets, thanks to this project.

12. FUTURE SCOPE

Having a technology that can translate hand sign language to its corresponding alphabet is a game changer in the field of communication and AI for the specially abled people such as deaf and dumb. With introduction of gesture recognition, the web app can easily be expanded to recognize letters beyond 'I', digits and other symbols plus gesture recognition can also allow controlling of software/hardware interfaces.

It will contribute to the development of improved communication for the deafened. The majority of people are unable to communicate via sign language, which creates a barrier to communication.

As a result, others will be able to learn and comprehend sign language and communicate with the deaf and dumb via the web app. According to scientific research, learning sign language improves cognitive abilities, attention span, and creativity.

13. APPENDIX

Source Code

Data_Collection_and_Data_Preprocessing (1).ipynb

```
# ***IMPORTING NECESSARY LIBRARIES***
from google.colab import drive
drive.mount('/content/drive')
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
!unzip '/content/drive/MyDrive/test_set.zip'
 !unzip '/content/drive/MyDrive/training_set.zip'
# **RENAMING DATA FILES**
def rename_imgs(file_name):
    folder_path = r'/content/drive/MyDrive/test_dataset'+file_name

    num = 0
    for file in os.listdir(folder_path):
        # if num%10 == 0:
        #     print(f'Renamed {num} files...')
        # os.rename(folder_path+'\\'+file,
folder_path+'\\'+file_name+'_'+str(num)+'.jpeg')
        num += 1
fn = ''
rename_imgs(fn)
file_names = ''
for fn in file_names:
    rename_imgs(fn)
```

```
# **DISPLAYING SAMPLE IMAGES FROM DATASET**
train_data_path = 'train_dataset/'
test_data_path = 'test_dataset/'
def display(img,sign=None):
    img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    fig = plt.figure(figsize=(7,7))
    ax = fig.add_subplot(111)
    plt.title(sign)
    ax.imshow(img)
## **Training Data Images**
sign_img = cv2.imread('/content/drive/MyDrive/train_dataset/3/23.jpg')
display(sign_img,'a')
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/A/244.jpg')
display(sign_img,'A')
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/D/7.jpg')
display(sign_img,'D')
sign_img = cv2.imread('/content/drive/MyDrive/train_dataset/3/340.jpg')
display(sign_img,'3')
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/H/7.jpg')
display(sign_img,'7')
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/Z/1.jpg')
display(sign_img,'Z')
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/P/8.jpg')
display(sign_img,'5')
# **Test Data Images**
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/S/15.jpg')
display(sign_img,'S')
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/Z/1.jpg')
display(sign_img,'Z')
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/7/8.jpg')
display(sign_img,'7')
```

AUGMENTATION AND PREPROCESSING THE DATASET
Creating ImageDataGenerator
```
image_gen = ImageDataGenerator(rotation_range=30,
                               width_shift_range=0.1,
                               height_shift_range=0.1,
                               shear_range=0.2,
                               zoom_range=0.2,
                               rescale=1/255,
                               horizontal_flip=True,
                               fill_mode='nearest',
                               validation_split=0.25)
```
Original Image
```
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/3/100.jpg')
display(sign_img,'3')
```
Augmented Images

```
display(image_gen.random_transform(sign_img))
display(image_gen.random_transform(sign_img))
```
SPLITING INTO TRAIN AND VALIDATION DATASET
Train Data Generator
```
train_data_gen =
image_gen.flow_from_directory('/content/drive/MyDrive/train_dataset',
                                target_size=(250,250),
                                batch_size=16,
                                shuffle=True,
                                class_mode='binary',
                                subset='training')
```
Validation Data Generator
```
validation_data_gen =
image_gen.flow_from_directory('/content/drive/MyDrive/train_dataset',
                                target_size=(250,250),
```

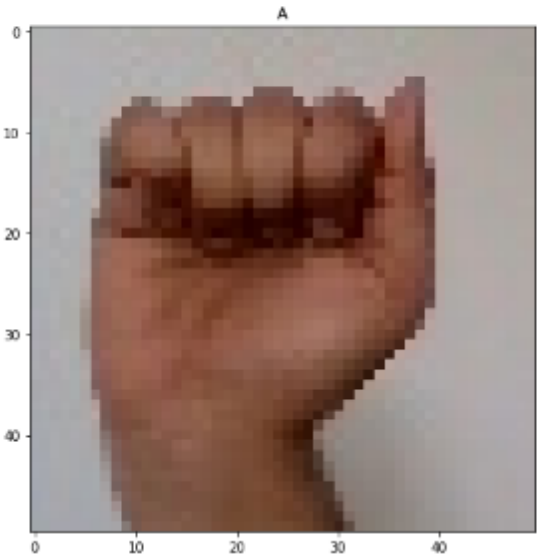
```
                                batch_size=16,
                                shuffle=True,
                                class_mode='binary',
                                subset='validation')
**Test Data Generator**
test_data_gen =
image_gen.flow_from_directory('/content/drive/MyDrive/test_dataset',
                                target_size=(250,250),
                                batch_size=8,
                                shuffle=True,
                                class_mode='categorical',
                                )
train_data_gen.class_indices
test_data_gen.classes
len(train_data_gen.classes)
len(test_data_gen.classes)
```

Output

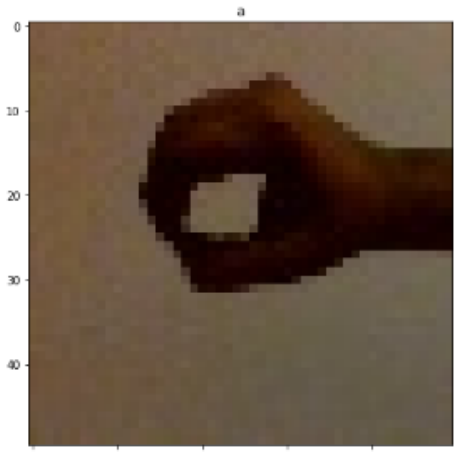
```
!unzip '/content/drive/MyDrive/training_set.zip'
```

```
extracting: training_set/G/1231.png
extracting: training_set/G/1232.png
 inflating: training_set/G/1233.png
 inflating: training_set/G/1234.png
 inflating: training_set/G/1235.png
 inflating: training_set/G/1236.png
 inflating: training_set/G/1237.png
 inflating: training_set/G/1238.png
 inflating: training_set/G/1239.png
 inflating: training_set/G/124.png
 inflating: training_set/G/1240.png
 inflating: training_set/G/1241.png
 inflating: training_set/G/1242.png
 inflating: training_set/G/1243.png
 inflating: training_set/G/1244.png
 inflating: training_set/G/1245.png
extracting: training_set/G/1246.png
 inflating: training_set/G/1247.png
 inflating: training_set/G/1248.png
 inflating: training_set/G/1249.png
 inflating: training_set/G/125.png
 inflating: training_set/G/1250.png
 inflating: training_set/G/1251.png
 inflating: training_set/G/1252.png
 inflating: training_set/G/1253.png
 inflating: training_set/G/1254.png
 inflating: training_set/G/1255.png
 inflating: training_set/G/1256.png
 inflating: training_set/G/1257.png
 inflating: training_set/G/1258.png
 inflating: training_set/G/1259.png
 inflating: training_set/G/126.png
 inflating: training_set/G/1260.png
 inflating: training_set/G/1261.png
extracting: training_set/G/1262.png
 inflating: training_set/G/1263.png
 inflating: training_set/G/1264.png
 inflating: training_set/G/1265.png
 inflating: training_set/G/1266.png
 inflating: training_set/G/1267.png
extracting: training_set/G/1268.png
 inflating: training_set/G/1269.png
 inflating: training_set/G/127.png
 inflating: training_set/G/1270.png
 inflating: training_set/G/1271.png
 inflating: training_set/G/1272.png
 inflating: training_set/G/1273.png
 inflating: training_set/G/1274.png
 inflating: training_set/G/1275.png
 inflating: training_set/G/1276.png
 inflating: training_set/G/1277.png
 inflating: training_set/G/1278.png
 inflating: training_set/G/1279.png
 inflating: training_set/G/128.png
 inflating: training_set/G/1280.png
 inflating: training_set/G/1281.png
 inflating: training_set/G/1282.png
 inflating: training_set/G/1283.png
 inflating: training_set/G/1284.png
```

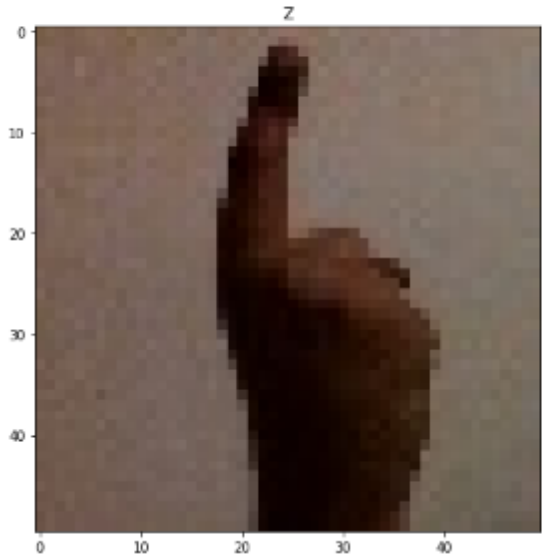
```
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/A/244.jpg')
display(sign_img,'A')
```




```
sign_img = cv2.imread('/content/drive/MyDrive/train_dataset/3/23.jpg')
display(sign_img,'a')
```

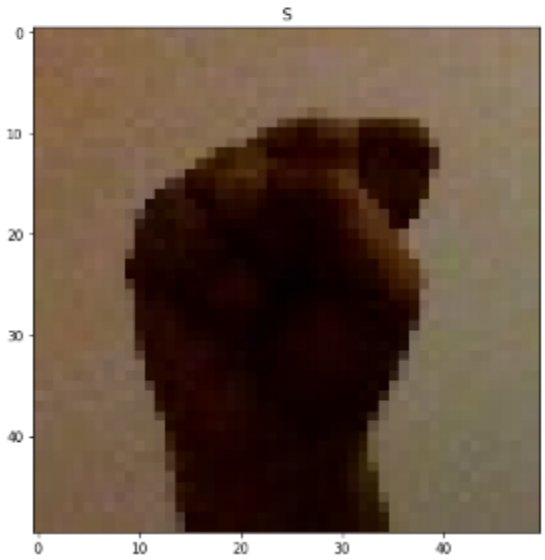


```
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/Z/1.jpg')
display(sign_img,'Z')
```



Test Data Images

```
sign_img = cv2.imread('/content/drive/MyDrive/test_dataset/S/15.jpg')
display(sign_img,'S')
```



Initialize the model.ipynb

#import imagedatagenerator

from keras.preprocessing.image import ImageDataGenerator

#training datagen

```
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=
0.2,horizontal_flip=True)
#testing datagen
test_datagen=ImageDataGenerator(rescale=1./255)

import tensorflow as tf
import os
# Initialize The Model
#create model
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
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt #to view graph in colab itself
import IPython.display as display
from PIL import Image
import pathlib
!unzip '/content/drive/MyDrive/test_set.zip'
! unzip '/content/drive/MyDrive/training_set.zip'
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,
64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),bat
```

```
ch_size=200,
                                    class_mode='categorical',color_mode="grayscale")
a=len(x_train)
b=len(x_test)
print(a)
print(b)
# Add Layers
#create model
model=Sequential()
```

Add the convolution layer.ipnb

```
#import imagedatagenerator
from keras.preprocessing.image import ImageDataGenerator
#training datagen
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=
0.2,horizontal_flip=True)
#testing datagen
test_datagen=ImageDataGenerator(rescale=1./255)
import tensorflow as tf
import os
# Initialize The Model
#create model
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
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt #to view graph in colab itself
import IPython.display as display
from PIL import Image
import pathlib
!unzip '/content/drive/MyDrive/test_set.zip'
!unzip '/content/drive/MyDrive/training_set.zip'
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
a=len(x_train)
b=len(x_test)
print(a)
print(b)
# Add Layers
#create model
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
```

Output:

```
[ ] x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,64),batch_size=200,
                                              class_mode='categorical',color_mode="grayscale")

    Found 15130 images belonging to 9 classes.

[ ] x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),batch_size=200,
                                            class_mode='categorical',color_mode="grayscale")

    Found 2250 images belonging to 9 classes.

[ ] a=len(x_train)
    b=len(x_test)

[ ] print(a)

    76

[ ] print(b)

    12
```

Add_the_flatten_layer.ipynb

```
#import imagedatagenerator
from keras.preprocessing.image import ImageDataGenerator
#training datagen
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=
0.2,horizontal_flip=True)
#testing datagen
test_datagen=ImageDataGenerator(rescale=1./255)
import tensorflow as tf
import os
# Initialize The Model
#create model
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
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt #to view graph in colab itself
import IPython.display as display
from PIL import Image
import pathlib
!unzip '/content/drive/MyDrive/test_set.zip'
!unzip '/content/drive/MyDrive/training_set.zip'
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,
64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),bat
ch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
a=len(x_train)
b=len(x_test)
print(a)
print(b)
# Add Layers
#create model
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
#1st hidden layer
model.add(Dense(units=512,activation='relu'))
#2nd hidden layer
model.add(Dense(units=261,activation='relu'))
#output layer
model.add(Dense(units=9,activation='softmax'))
```

Output:

Add Layers

```
[ ] #create model
    model=Sequential()
```

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

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

```
[ ] model.add(Flatten())
```

```
[ ] #1st hidden layer
    model.add(Dense(units=512,activation='relu'))
    #2nd hidden layer
    model.add(Dense(units=261,activation='relu'))
```

```
[ ] #output layer
    model.add(Dense(units=9,activation='softmax'))
```

Adding_The_Dense_Layers.ipynb

#import imagedatagenerator
from keras.preprocessing.image import ImageDataGenerator
#training datagen
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
#testing datagen
test_datagen=ImageDataGenerator(rescale=1./255)
import tensorflow as tf
import os
# Initialize The Model
#create model
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
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt #to view graph in colab itself
import IPython.display as display
from PIL import Image
import pathlib
!unzip '/content/drive/MyDrive/test_set.zip'
!unzip '/content/drive/MyDrive/training_set.zip'
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,
64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),bat
ch_size=200,
                                        class_mode='categorical',color_mode="grayscale")
a=len(x_train)
b=len(x_test)
print(a)
print(b)
# Add Layers
#create model
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
#1st hidden layer
model.add(Dense(units=512,activation='relu'))
#2nd hidden layer
model.add(Dense(units=261,activation='relu'))
#output layer
model.add(Dense(units=9,activation='softmax'))
```

Compile_The_Model.ipynb

```
#import imagedatagenerator
from keras.preprocessing.image import ImageDataGenerator
#training datagen
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=
0.2,horizontal_flip=True)
#testing datagen
test_datagen=ImageDataGenerator(rescale=1./255)
import tensorflow as tf
import os
# Initialize The Model
#create model
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
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt #to view graph in colab itself
import IPython.display as display
from PIL import Image
import pathlib
!unzip '/content/test_set.zip'
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,
64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),bat
ch_size=200,
```

```
                                    class_mode='categorical',color_mode="grayscale")
a=len(x_train)
b=len(x_test)
print(a)
print(b)
# Add Layers
#create model
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
#1st hidden layer
model.add(Dense(units=512,activation='relu'))
#2nd hidden layer
model.add(Dense(units=261,activation='relu'))
#output layer
model.add(Dense(units=9,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accurac
y'])
```

Fit_And_Save_The_Model.ipynb

```
#import imagedatagenerator
from keras.preprocessing.image import ImageDataGenerator
#training datagen
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=
0.2,horizontal_flip=True)
#testing datagen
test_datagen=ImageDataGenerator(rescale=1./255)
import tensorflow as tf
import os
# Initialize The Model
```

```
#create model
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
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt #to view graph in colab itself
import IPython.display as display
from PIL import Image
import pathlib
from google.colab import drive
drive.mount('/content/drive')
!unzip '/content/drive/MyDrive/test_set.zip'
!unzip '/content/drive/MyDrive/training_set.zip'
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,
64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),bat
ch_size=200,
                                        class_mode='categorical',color_mode="grayscale")
a=len(x_train)
b=len(x_test)
print(a)
print(b)
# Add Layers
#create model
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
```

model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
#1st hidden layer
model.add(Dense(units=512,activation='relu'))
#2nd hidden layer
model.add(Dense(units=261,activation='relu'))
#output layer
model.add(Dense(units=9,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
model.save('aslpng2.h5')

Output:

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

] model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  """Entry point for launching an IPython kernel.
Epoch 1/10
76/76 [==============================] - 84s 1s/step - loss: 0.4655 - accuracy: 0.8551 - val_loss: 0.3218 - val_accuracy: 0.9529
Epoch 2/10
76/76 [==============================] - 83s 1s/step - loss: 0.0470 - accuracy: 0.9866 - val_loss: 0.2929 - val_accuracy: 0.9667
Epoch 3/10
76/76 [==============================] - 84s 1s/step - loss: 0.0170 - accuracy: 0.9959 - val_loss: 0.2685 - val_accuracy: 0.9760
Epoch 4/10
76/76 [==============================] - 85s 1s/step - loss: 0.0101 - accuracy: 0.9973 - val_loss: 0.2647 - val_accuracy: 0.9756
Epoch 5/10
76/76 [==============================] - 82s 1s/step - loss: 0.0065 - accuracy: 0.9984 - val_loss: 0.2926 - val_accuracy: 0.9680
Epoch 6/10
76/76 [==============================] - 84s 1s/step - loss: 0.0094 - accuracy: 0.9976 - val_loss: 0.2773 - val_accuracy: 0.9756
Epoch 7/10
76/76 [==============================] - 84s 1s/step - loss: 0.0039 - accuracy: 0.9991 - val_loss: 0.3409 - val_accuracy: 0.9751
Epoch 8/10
76/76 [==============================] - 82s 1s/step - loss: 0.0106 - accuracy: 0.9972 - val_loss: 0.3067 - val_accuracy: 0.9760
Epoch 9/10
76/76 [==============================] - 84s 1s/step - loss: 0.0037 - accuracy: 0.9989 - val_loss: 0.3383 - val_accuracy: 0.9760
Epoch 10/10
76/76 [==============================] - 86s 1s/step - loss: 0.0013 - accuracy: 0.9999 - val_loss: 0.3545 - val_accuracy: 0.9764
<keras.callbacks.History at 0x7f2c805f1390>

] model.save('aslpng2.h5')
```

Import_ImageDataGenerator_Library_And_Configure_It.ipynb

# **Image Preprocessing**
# **Import ImageDataGenerator Library And Configure It**
from tensorflow.keras.preprocessing.image import ImageDataGenerator
# Training Datagen
train_datagen =
ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical

```
_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
import tensorflow as tf
import os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout,
MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import IPython.display as display
from PIL import Image
import pathlib
```

Import_The_Required_Model_ Building_Libraries.ipynb

```
#import imagedatagenerator
from keras.preprocessing.image import ImageDataGenerator
#training datagen
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=
0.2,horizontal_flip=True)
#testing datagen
test_datagen=ImageDataGenerator(rescale=1./255)
import tensorflow as tf
import os
# IMPORTING LIBRARIES TO INITIALIZE NEURAL NETWORK LAYER
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
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt #to view graph in colab itself
import IPython.display as display
from PIL import Image
import pathlib
!unzip '/content/drive/MyDrive/test_set.zip'
!unzip '/content/drive/MyDrive/training_set.zip'
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,
64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),bat
ch_size=200,
                                        class_mode='categorical',color_mode="grayscale")
a=len(x_train)
b=len(x_test)
print(a)
print(b)
```

Test the model.ipynb

```
!unzip '/content/drive/MyDrive/test_set.zip'
!unzip '/content/drive/MyDrive/training_set.zip'
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import cv2
img = image.load_img('/content/test_set/H/107.png',target_size = (100,100))
img
from skimage.transform import resize
def detect(frame):
```

```
    img=image.img_to_array(frame)
    img = resize(img,(64,64,1))
    img = np.expand_dims(img,axis=0)
    pred=np.argmax(model.predict(img))
    op=['A','B','C','D','E','F','G','H','I']
    print("THE PREDICTED LETTER IS ",op[pred])
import model
model=load_model('/content/drive/MyDrive/test_dataset/H/107.png')
img=image.load_img('/content/drive/MyDrive/test_dataset/H/107.png')
detect(img)
img = image.load_img('/content/test_set/A/110.png')
pred=detect(img)
```

Output:

TEST THE MODEL

```
[ ] from tensorflow.keras.models import load_model
    from tensorflow.keras.preprocessing import image
    import numpy as np
    import cv2
```

```
[ ] from skimage.transform import resize
    def detect(frame):
        img=image.img_to_array(frame)
        img = resize(img,(64,64,1))
        img = np.expand_dims(img,axis=0)
        pred=np.argmax(model.predict(img))
        op=['A','B','C','D','E','F','G','H','I']
        print("THE PREDICTED LETTER IS ",op[pred])
```

```
[ ] img=image.load_img("/content/test_set/E/107.png")
    detect(img)

    1/1 [==============================] - 0s 24ms/step
    THE PREDICTED LETTER IS  E
```

```
[ ] img = image.load_img('/content/test_set/H/110.png')
    pred=detect(img)

    1/1 [==============================] - 0s 26ms/step
    THE PREDICTED LETTER IS  H
```

```
[ ] img=image.load_img('/content/test_set/D/111.png')
    detect(img)

    1/1 [==============================] - 0s 180ms/step
    THE PREDICTED LETTER IS  D
```

Open.cv

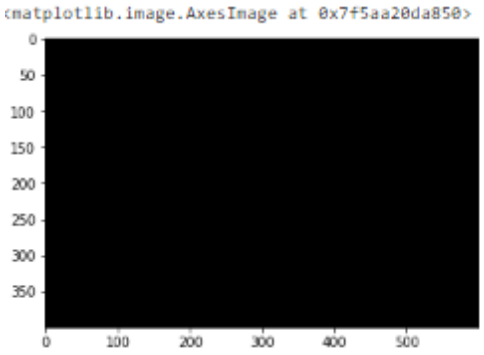
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
# **Image processing**
# Create a image

img1 = np.zeros((400,600,3),np.uint8)
plt.imshow(img1)
# Draw a circle

circle = cv2.circle(img1, (300,200), 50, (255,0,0), -1)   # (0,0,0)--->(R,G,B)
plt.imshow(img1)

# Drawing rectangle

rectangle = cv2.rectangle(img1,(200,100),(400,300),(0,255,0),6)
plt.imshow(img1)

line1 = cv2.line(img1,(200,100),(400,300),(0,0,255),4)
line2 = cv2.line(img1,(200,300),(400,100),(0,0,255),4)
plt.imshow(img1)

circle = cv2.circle(img1, (300,200), 50, (255,255,0), -1)   # (0,0,0)--->(R,G,B)
plt.imshow(img1)
# Text on image

text = cv2.putText(img1, 'openCV', (200,50), cv2.FONT_HERSHEY_SIMPLEX,
2, (255,255,255),5)
plt.imshow(img1)
```

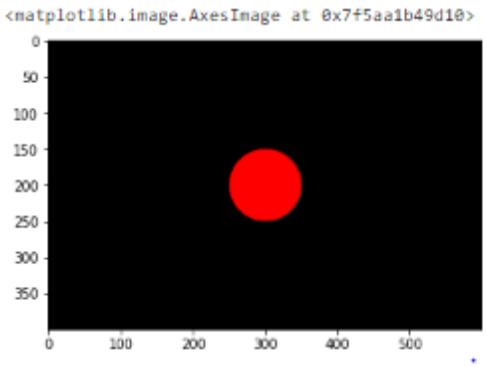
```
# Reading the image

img = cv2.imread('/content/boy.png',1)
plt.imshow(img)
# Convert BGR to RGB

img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img_rgb)
# Convert BGR to Gray

img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(img_gray)
# Finding shape

img_rgb.shape
img_gray.shape
# Resize the image

resize = cv2.resize(img_rgb,(500,1000))
print(resize.shape)
plt.imshow(resize)
# Image crop

crop = resize[130:370,150:300]
plt.imshow(crop)

# Edge Detection

edge = cv2.Canny(img_rgb,100,200)
plt.imshow(edge)
```

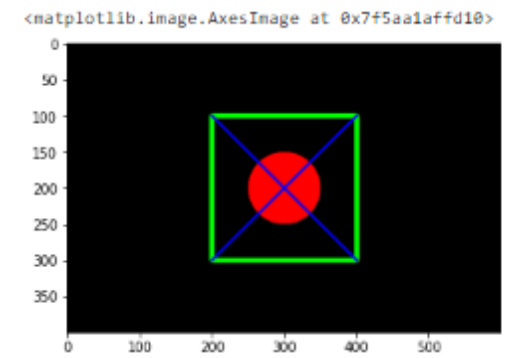
Blur image

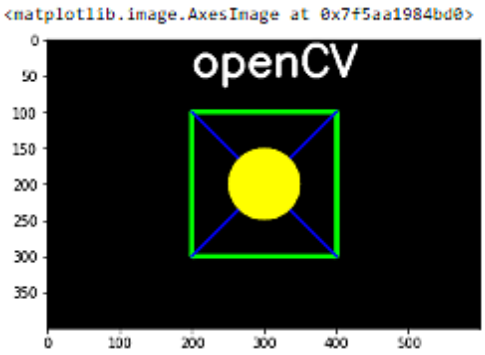
r = resize[130:370,150:300]
blur = cv2.GaussianBlur(r,(13,13),cv2.BORDER_DEFAULT)
plt.imshow(resize)
plt.imshow(blur)

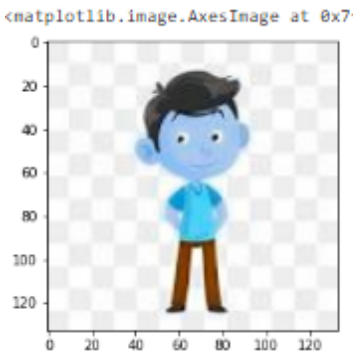
Output:

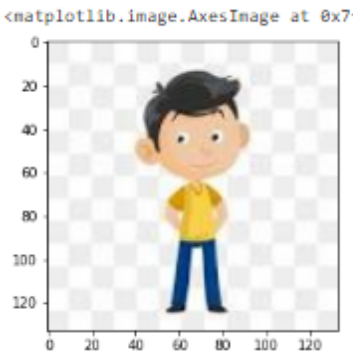


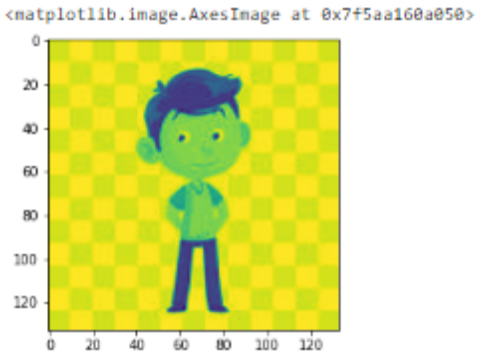


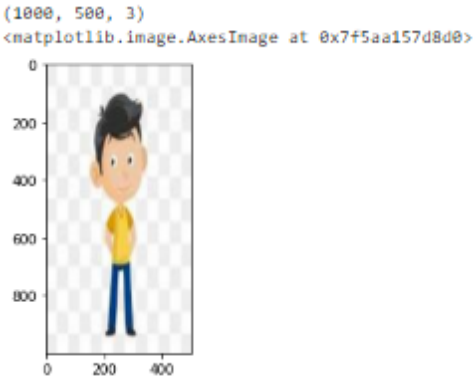


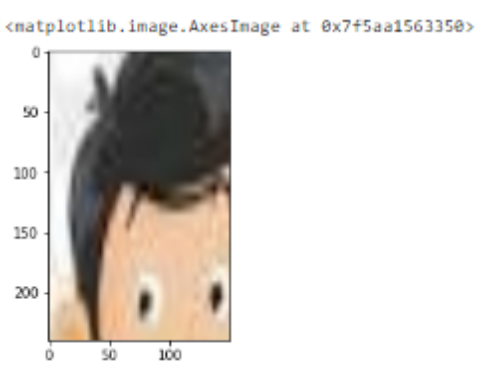


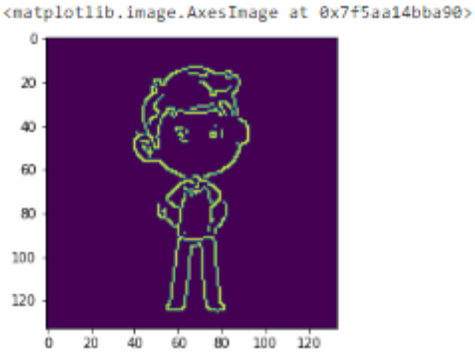












Index.html

```
<!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>SmartBridge_WebApp_VideoTemplate</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="assets/css/Banner-Heading-Image.css">
    <link rel="stylesheet" href="assets/css/Navbar-Centered-Brand.css">
```

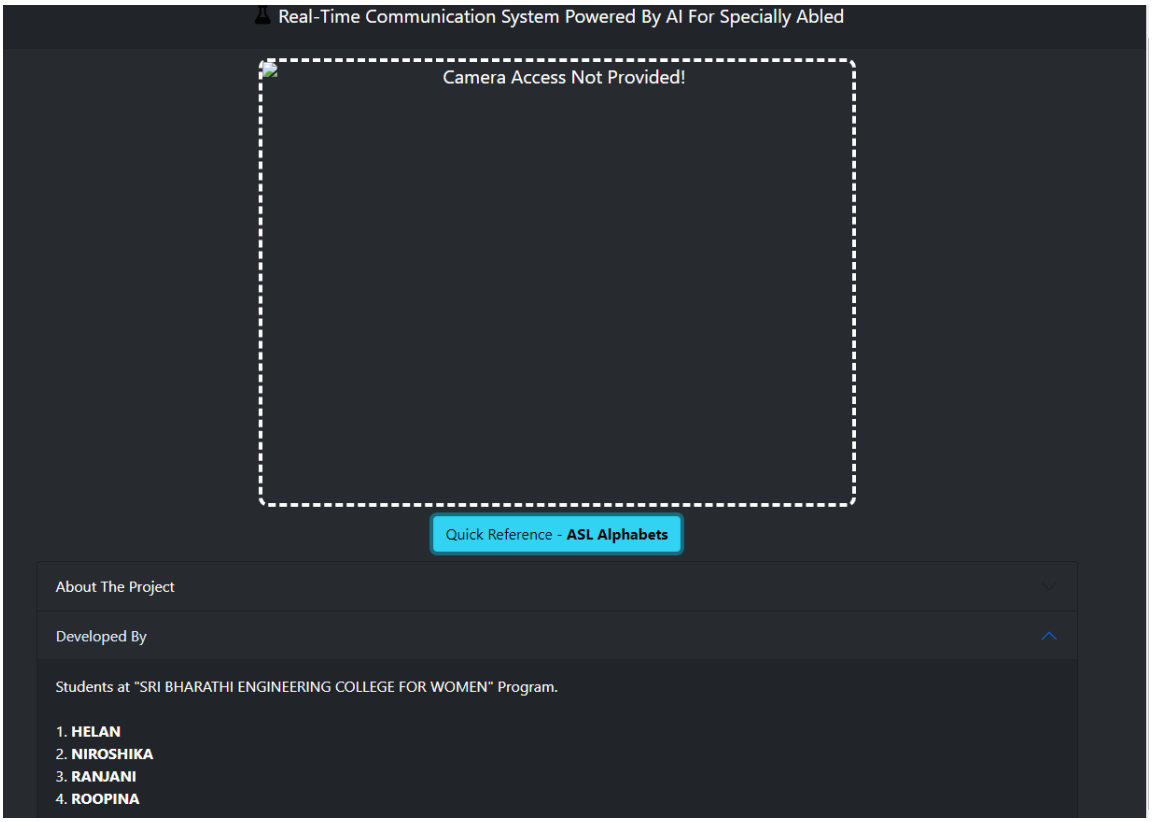
```
    <link rel="stylesheet" href="assets/css/styles.css">
</head>

<body style="background: rgb(39,43,48);">
    <nav class="navbar navbar-light navbar-expand-md py-3" style="background: #212529;">
        <div class="container">
                    <div></div><a class="navbar-brand d-flex align-items-center" href="#"><span
                        class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex justify-content-center align-items-center me-2 bs-icon"><i
                                    class="fas fa-flask"></i></span><span style="color: rgb(255,255,255);">Real-Time Communication
                System Powered By AI&amp;nbsp;For Specially Abled</span></a>
            <div></div>
        </div>
    </nav>
    <section>
        <div class="d-flex flex-column justify-content-center align-items-center">
            <div class="d-flex flex-column justify-content-center align-items-center" id="div-video-feed"
                        style="width: 640px;height: 480px;margin: 10px;min-height: 480px;min-width: 640px;border-radius: 10px;border: 4px dashed rgb(255,255,255);">
                    <img src="{{ url_for('video_feed') }}" style="width: 100%;height: 100%;color: rgb(255,255,255);text-align: center;font-size: 20px;"
                alt="Camera Access Not Provided!">
            </div>
        </div>
            <div class="d-flex flex-column justify-content-center align-items-center" style="margin-bottom: 10px;"><button
```

```
                class="btn btn-info" type="button" data-bs-target="#modal-1" data-bs-toggle="modal">Quick Reference
            -<strong> ASL Alphabets</strong></button></div>
    </section>
    <section>
        <div class="container">
            <div class="accordion text-white" role="tablist" id="accordion-1">
                <div class="accordion-item" style="background: rgb(33,37,41);">
                    <h2 class="accordion-header" role="tab"><button class="accordion-button" data-bs-toggle="collapse"
                        data-bs-target="#accordion-1 .item-1" aria-expanded="true"
                        aria-controls="accordion-1 .item-1"
                                                    style="background:  rgb(39,43,48);color: rgb(255,255,255);">About The Project</button></h2>
                <div class="accordion-collapse collapse show item-1" role="tabpanel" data-bs-parent="#accordion-1">
                    <div class="accordion-body">
                        <p class="mb-0">Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human
                            intelligence, such as speech and voice recognition, visual perception, predictive text
                            functionality, decision-making, and a variety of other tasks, AI can assist people with
                            disabilities by significantly improving their ability to get around and participate in
                            daily activities.<br><br>Currently, Sign Recognition is available <strong>only for
                            alphabets A-I</strong> and not for J-Z, since J-Z alphabets also require Gesture
                            Recognition for them to be able to be predicted correctly to a
```

```
certain degree of
                        accuracy.</p>
                    </div>
                </div>
            </div>
            <div class="accordion-item" style="background: rgb(33,37,41);">
                <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="background:  rgb(39,43,48);color: rgb(231,241,255);">Developed By</button></h2>
                <div class="accordion-collapse collapse item-2" role="tabpanel" data-bs-parent="#accordion-1">
                    <div class="accordion-body">
                                        <p class="mb-0">Students at "SRI BHARATHI ENGINEERING COLLEGE FOR WOMEN"
                        Program.<br><br>1. <strong>HELAN</strong><br>2.
                            <strong>NIROSHIKA</strong> <br>3. <strong>RANJANI </strong><br>4. <strong>ROOPINA</strong>
                        </p>
                    </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"><img src="{{ url_for('static', filename='img/ASL_Alphabets') }}" width="100%"></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>
```

Output:



Banner-Heading-Image.css

```
.fit-cover {
  object-fit: cover;
}
```

Navbar-Centered-Brand.css

```
.bs-icon {
  --bs-icon-size: .75rem;
  display: flex;
  flex-shrink: 0;
  justify-content: center;
```

app.py

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

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

camera.py

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

```
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('asl_model.h5') # Execute Local Trained Model
         # self.model = load_model('IBM_Communication_Model.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 = image.load_img('image.jpg', target_size=(64,64))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
```

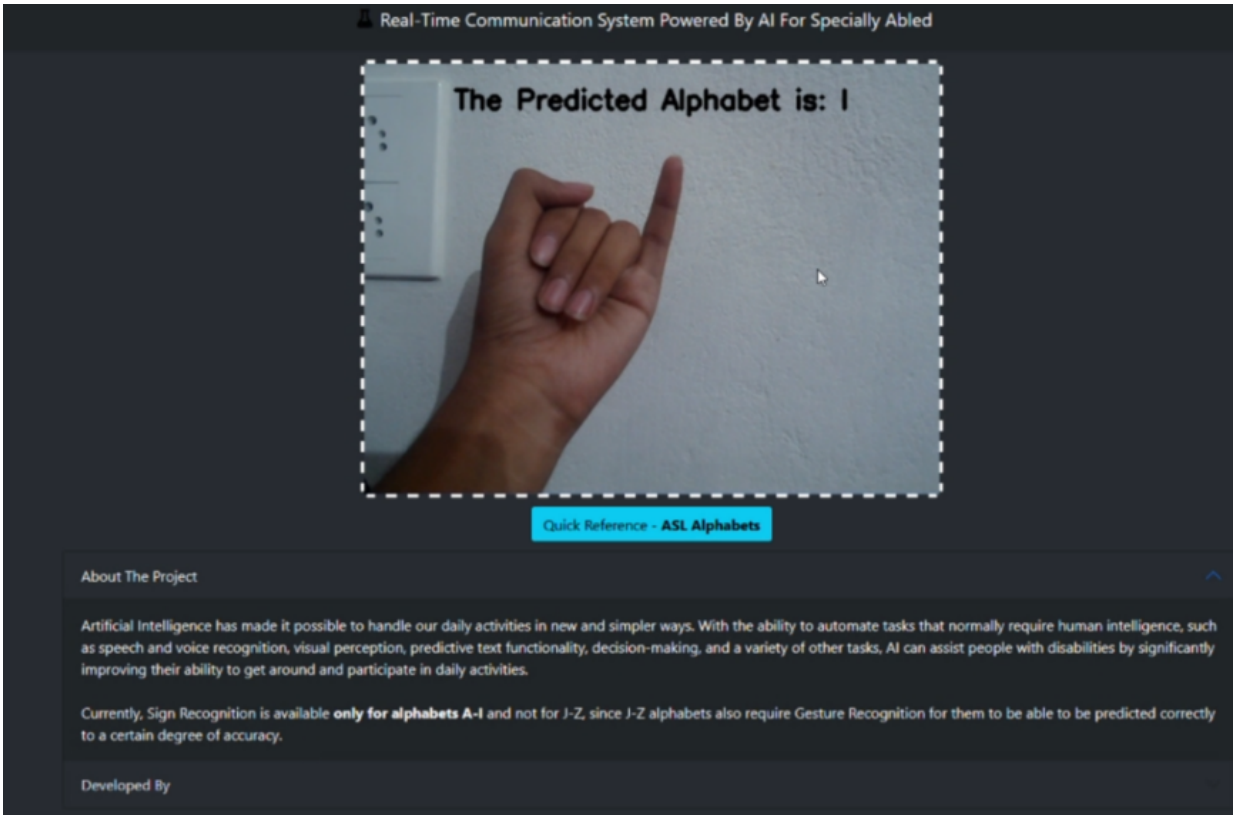
```
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
                                cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

flask1.py

```
from flask import Flask, render_template
app = Flask(__name__, template_folder='Templates', static_folder='static')
@app.route('/')
def index():
    return render_template('index.html')
if __name__=='__main__':
    a pp.run(debug = True)
```

Output:


Real-Time Communication System Powered By AI For Specially Abled
The Predicted Alphabet is: F
Quick Reference - ASL Alphabets
About The Project
Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly improving their ability to get around and participate in daily activities.
Currently, Sign Recognition is available only for alphabets A-I and not for J-Z, since J-Z alphabets also require Gesture Recognition for them to be able to be predicted correctly to a certain degree of accuracy.
Developed By


Real-Time Communication System Powered By AI For Specially Abled
The Predicted Alphabet is: I
Quick Reference - ASL Alphabets
About The Project
Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly improving their ability to get around and participate in daily activities.
Currently, Sign Recognition is available only for alphabets A-I and not for J-Z, since J-Z alphabets also require Gesture Recognition for them to be able to be predicted correctly to a certain degree of accuracy.
Developed By

GitHub & Project Demo Link

Github

https://github.com/IBM-EPBL/IBM-Project-6465-1658829715

Project Demo Link

https://drive.google.com/file/d/1bB2Qlm_ARRJaIilv3pc3hLe6u-50le8i/view?usp=share_link