Real-Time Communication System Powered by AI for Specially Abled

DEPARTMENT OF INFORMATION TECHNOLOGY PANIMALAR ENGINEERING COLLEGE, ,CHENNAi-6000123

Team ID: PNT2022TMID01421

Team Leader - Yashwanth M

Team Members - K Raj Kumar ,

P Vettri Chezhian,

M Naveen Kumar

Project Report Index

ABSTRACT

People with physical limitations that they often encounter in their lives. Deaf and speech-impaired people have limited communication. Any normal person would see the situation, listen, and speak up to respond. However, there are those who are unlucky who have been deprived of this precious gift. This creates a gap between the common people and the underprivileged. This application helps both communicate with each other. This system mainly he consists of two modules. The first module extracts Indian Sign Language (ISL) gestures from real-time video and maps them to human-understandable audio. So the second module takes natural language as input and maps it to corresponding animated Indian Sign Language gestures..

1. INTRODUCTION

1.1 Project Overview

One of the most precious gifts that nature has given mankind is the ability to express ourselves in response to the events around us. This project aims to support the deaf and mute by creating a new system that helps convert sign language to text and speech to facilitate communication with the audience. The system consists of a gesture recognition algorithm that converts gestures into letters or numbers. These electrical signals are processed using a computer and Python-based backend for text-to-speech conversion. The system has his two modes of operation: phrase recall mode and letter recall mode. Phrase recall mode pronounces words at once, letter recall mode pronounces individual letters. This project forms a basic infrastructure that can later be extended to include various sign languages.

1.2 Purpose

Most of the tasks we do in our daily lives are speaking and listening. People who are deaf or mute have difficulty communicating with people who cannot understand or misinterpret sign language. The idea of this project is to use technological advances to build bridges and enable disadvantaged people to participate in life on an equal footing, achieve their goals, and break down or overcome challenges and barriers. In this research, we create an Android-based application that can directly interpret sign language presented in written form by deaf people. The translation process starts with detecting hands in OpenCV and translating the signal from the K-NN classification.

2. LITERATURE SURVEY

Numerous inventions of the 20th and 21st centuries have changed and advanced our perspectives and traditional approaches to the world. Technological advances in both software and hardware have given rise to different versions and methods of sign language interpretation. Few projects have been successful, but the larger scale of these implemented projects has been impractical, with various glitches and bugs depending on the individual project, as well as time lags and database limitations. did.

2.1. Existing problem

Software that helps people rely on various criteria and factors to function smoothly. Existing systems must be connected to the Internet. This can vary greatly by location and by device version, hardware, software, and various other variables involved. The consumer market is diverse, owning different brands and versions of smartphones with different combinations of hardware and software specifications, so the software must be glitch-free and compatible with all devices, which is very It can represent complex and diverse possibilities.

2.2. References

PAPER DETAILS	ABSTRACT	EXPLAINATION
Design of a Communication System using Sign Language aid for Differently Abled Peoples.	One of the most precious gifts of nature to the human race is the ability to express itself by responding to the events that occur in its environment. Every normal person sees, hears, and then reacts to the situations by expressing himself. But some less lucky ones are deprived of this precious gift. Such people, especially deaf and mute, rely on some sort of gesture language to communicate their feelings to others. The deaf, dumb and the blind follow similar problems when it comes to the use of computers. In the era of advanced technologies, where computers, laptops and other processor-based devices are an integral part of everyday life, efforts must be made to make the disabilities in life more independent.	This software uses sign language recognition along with text to speech software to interpret The message.
Smart Communication for Differently Abled People	Our day to day life most of the task we carry out involves speaking and hearing. The deaf and dumb people have difficulty in communicating with others who cannot understand sign language and mis-interpreter. In this paper, we designed a simple Embedded System based device for solving this problem. We have used flex sensor for getting the data from the deaf and dumb using sign language. When deaf wants to convey any messages then the user will give his voice as input to the android based voice app. Then the app will transfer this particular speech in to text and it will displayed in LCD. For Dumb People if they want to convey any messages to user Two Flex sensors are used to play voice. For Blind People, if they want to read any books or text the camera will act as eye to capture the text region and using Tesseract it will convert in to voice.	We have used flex sensor for getting the data from the deaf and dumb using sign language. When deaf wants to convey any messages then the user will give his voice as input to an android based voice app. For Dumb People if they want to read any books or text the camera will act as eye to capture the text region and using Tesseract it will convert in to voice.
Real-time Communication System for the Deaf and Dumb	This project aims to aid the deaf-mute by creation of a new system that helps convert sign language to text and speech for easier communication with audience. The system consists of a gesture recognizer hand-glove which converts gestures into electrical signals using flex sensors. These electrical signals are then processed using an Arduino microcontroller and a Python-based backend for text-to-speech conversion. The glove includes two modes of operation – phrase fetch mode and letter fetch mode. The phrase fetch mode speaks out words at once, while the letter fetch mode speaks out individual letters. This project forms a base infrastructure which can later be augmented with addition of different Sign Languages.	This project aims to aid the deaf-mute by creation of a new system that helps convert sign language to text and speech. System consists of a gesture recognizer hand-glove which converts gestures into electrical signals using flex sensors. The glove includes two modes of operation – phrase fetch mode and letter fetch mode.
Two Way Communicator between Deaf and Dumb People and Normal People	This system consists mainly of two modules, the first module is Indian Sign Language (ISL) gestures from real-time video and mapping it with human-Understandable speech. Accordingly, the second module is the natural language as Input and card with equivalent Indian Sign Language animated gestures.	ISL is an attempt to teach computers how to use gestures from real-time video and mapping them to human speech.

Sign Language Recognition System to aid Deaf-dumb People Using PCA.	This paper presents design and implementation of real-time sign language recognition system, to 26 gestures from the Indian sign language with MATLAB.	Real-time sign language recognition system to 26 gestures from the Indian sign language has been developed and tested in MATLAB, an open source software platform for developing computer programs with built-in sign language interpreters at Microsoft Research.
Sign Language to Text and Vice Versa Recognition using Computer Vision in Marathi.	In this system edge detection algorithm is used to recognize the input character image gray scale and recognition of the edges of the hand gesture. The system is able to handle the different input records images of alphabets, words, sentences, and translates them in text and vice versa. The system is designed to translate the Marathi sign language to text.	The system is able to handle the different input records images of alphabets, words, sentences, and translates them in text. System is designed to translate the Marathi sign language to text. Edge detection algorithm is used to recognize the input character image gray scale and recognition of the edges of the hand gesture.
Sign Language Learning based on Android for Deaf and Speech Impaired People.	This research makes an Android-based application that can directly interpret Sign language presented by deaf people in written language. Translation process Starts with the detection of hands with OpenCV and translation of and signals The K-NN classification. Tutorial features added in this application with the goal to train intensively to guide the user when using the sign language.	This research makes an Android-based application that can directly interpret Sign language presented by deaf people in written language. Tutorial features added in this application with the goal to train intensively to guide the user when using the sign language. The translation process starts with the detection of hands with OpenCV and translation of and signals.
SOFTWARE ASSISTANCE TO DEAF AND DUMB USING HANDSHAPE ALGORITHM	This application helps the deaf and dumb person to communicate with the rest of the world using sign language communication plays an important role for human beings. Communication is treated as a life skill. Keeping these important words in mind we present this project to mainly focus on aiding the speech impaired and paralyzed patients. Our work helps in improving the communication with the deaf and dumb.	This application helps the deaf and dumb person to communicate with the rest of the world using sign language communication plays an important role for human beings. We mainly focus on aiding the speech impaired and paralyzed patients. Our work helps in improving the communication with the deaf and dumb.
COMMUNICATION DEVICE FOR DIFFERENTLY ABLED PEOPLE:A PROTOTYPE MODEL	The process of communication between marginalized communities like deaf-blind-dumb people has always been a matter of great concern and these differently abled people are not able to easily communicate their thoughts and talks with other people as normal people does by using mobile phones, etc. So, it is the greatest need of this hour to think and act upon the development of such people as they are also the equal part of our society.	The prototype software built here provides various methods to interpret hand signs and uses sign to speech to help the user.

2.3. Problem Statement Definition

Communication between the specially-abled and the ordinary people has always been a challenging task. Especially, the way of communication between people with disabilities cannot be easily learned by Ordinary people.

Since communication needs to be faster and accurate. This system should also covey the message in case of emergency. Moreover, the message should be transferred from one person to another without any change in the content

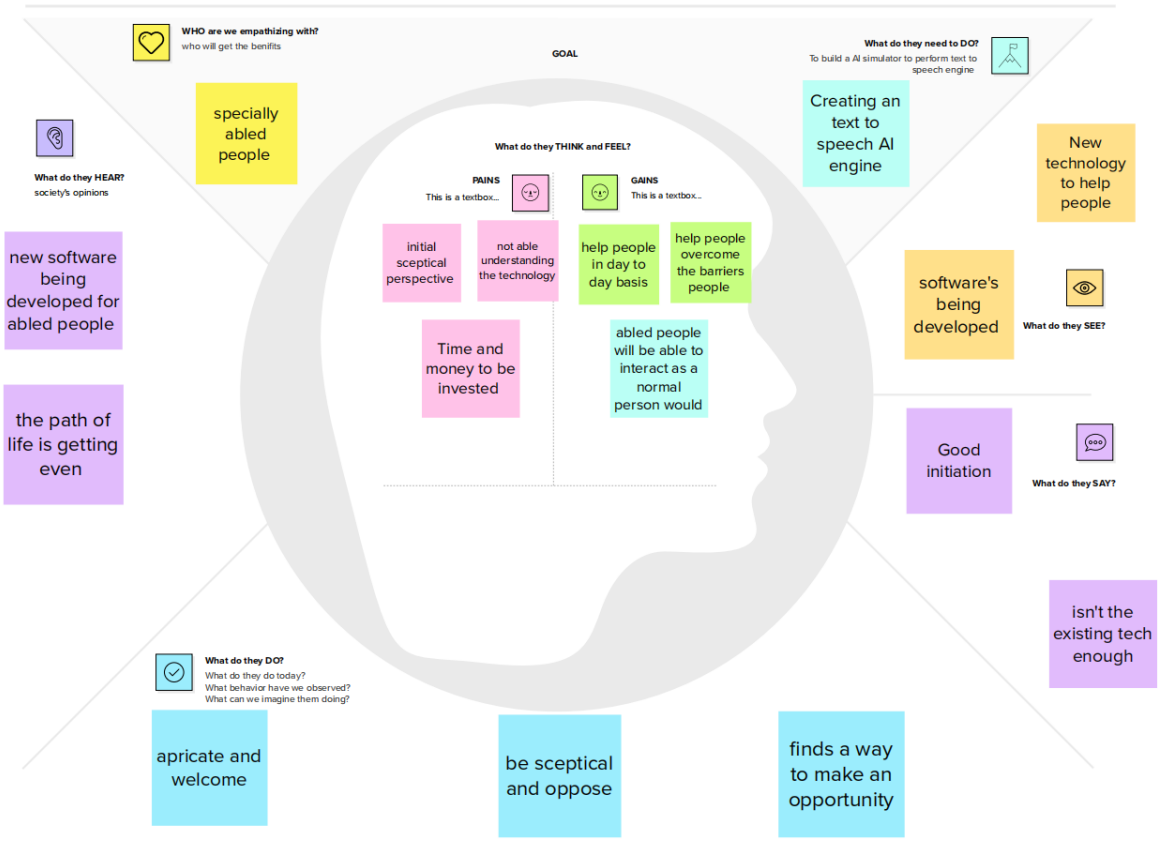
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	An ordinary person	Express my thoughts with a specially-abled person.	I don't know sign language	I didn't learned sign language	Frustrated.
PS-2	Specially abled person	Communicate with an ordinary person	it's difficultto express	The ordinary person doesn't know sign language	Frustrated.

3 . IDEATION & PROPOSED SOLUTION

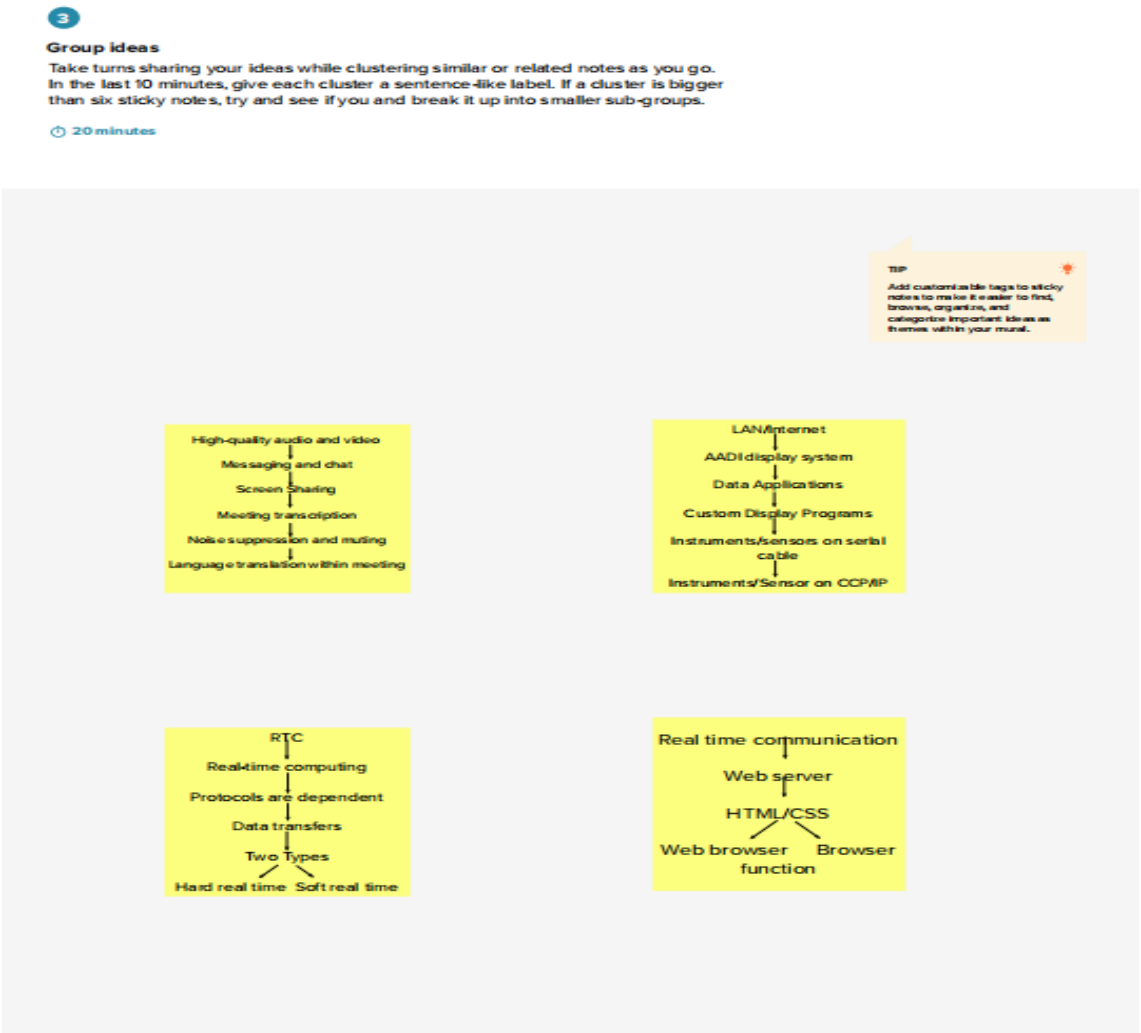
The basic idea of this project is to interpret hand signals, or sign language, decode them using a database model, analyze them, and potentially infer words or characters presented in front of a camera. The highest is to guess. This work involves a camera tracking any visible finger or hand gestures and comparing these characters to a database. An algorithm within the system, implemented via Python code, then matches the gesture of the most similar character, and the algorithm sends the character to the other party. The next step is to implement text-to-speech. That is, decoded hand signals are converted to text and text-to-speech. The proposed solution includes his python code as backend, camera as media and application as functional basis.

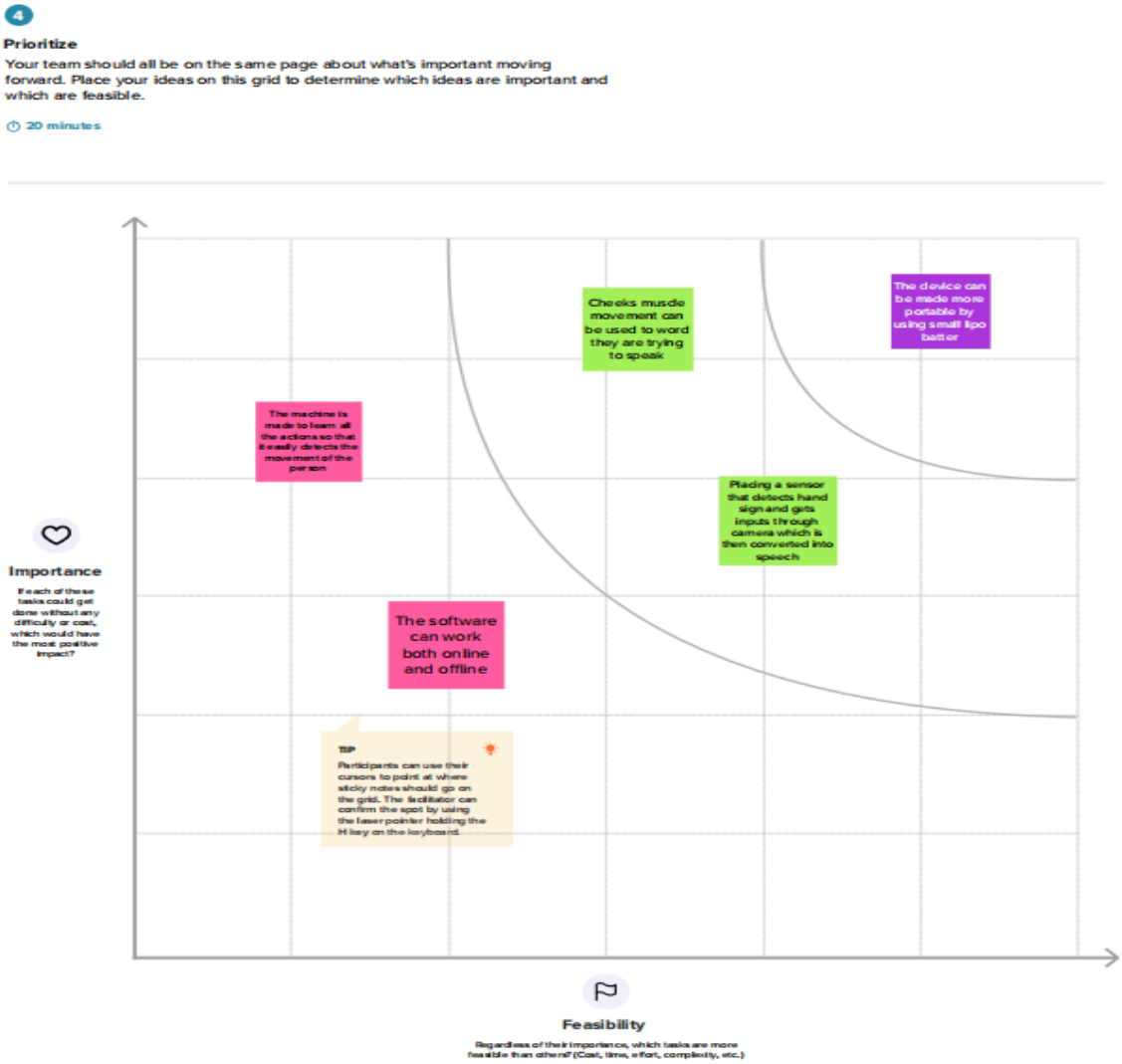
3.1. Empathy Map Canvas

In this activity you are expected to prepare the empathy map canvas to capture the user Pains & Gains, Prepare list of problem statements.



3.2 Ideation & Brainstorming





3.3 Proposed Solution

The proposed solution interprets sign language, decodes it, converts it to text, and then converts the text to speech. The final output is delivered audibly for the general public to understand and be efficient with. A computer acts as the medium of interpretation, and a Python file processes the data in the backend, followed by a text-to-speech algorithm that converts the text to speech. This entire process is accessed through an application interface.

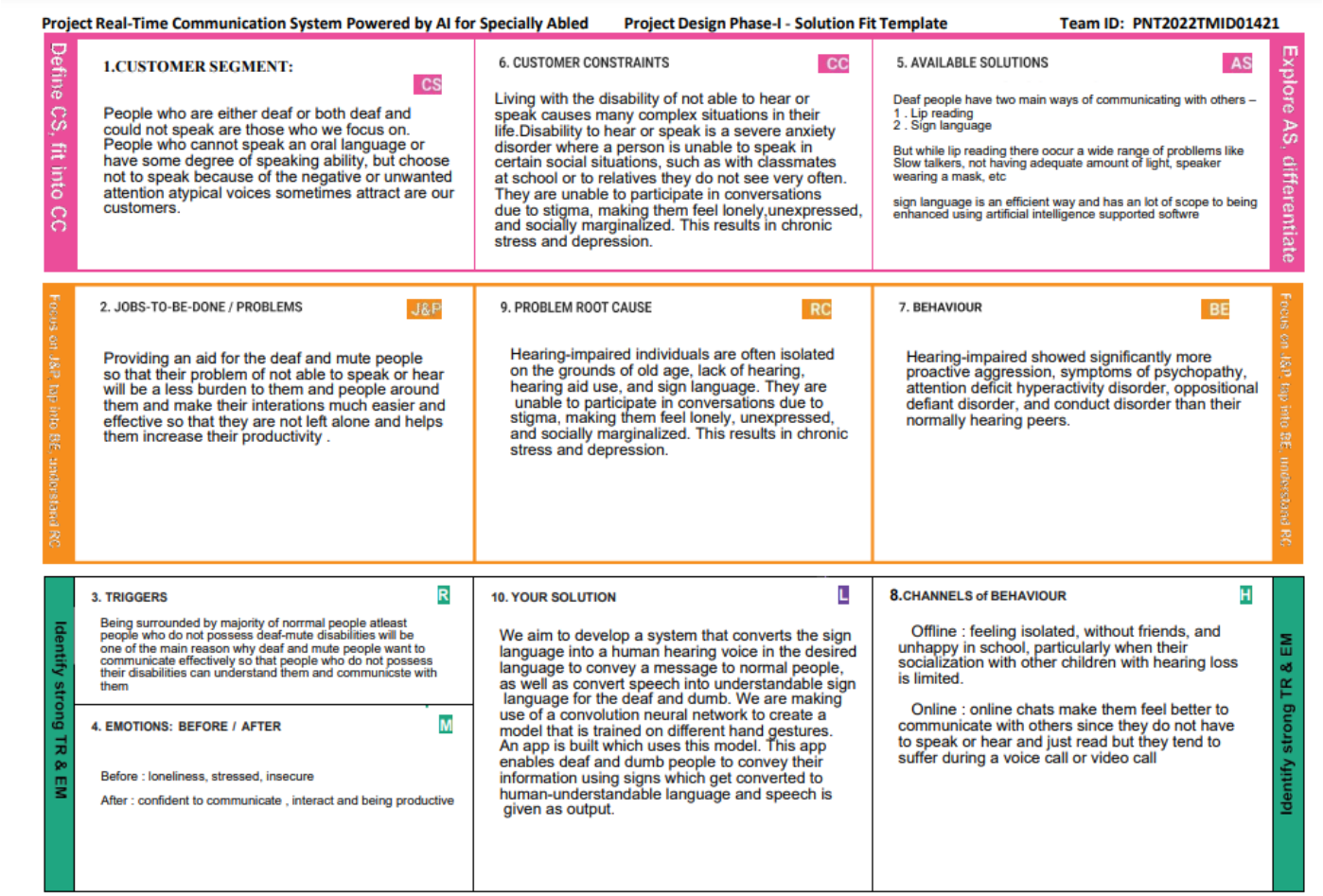
S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people since normal people are not trained on hand sign language in emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like

		speech cannot be used. Making communication between deaf-mute people and normal person effective is the problem to solve here.
2	Idea / Solution description	We aim to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.
3	Novelty / Uniqueness	1. Since deaf-mute people are usually deprived of normal communication with other people, they have to rely on an interpreter or some visual communication. Now the interpreter can not be available always, so this project can help eliminate the dependency on the interpreter. 2. A web based version of the application will increase the reach to more people. 3.Integrating hand gesture recognition system using computer vision for establishing 2-way communication system.
4	Social Impact / Customer Satisfaction	Sign language is a visual language that is used by deaf people as their mother tongue. Unlike acoustically conveyed sound patterns, sign language uses body language and manual communication to fluidly convey the thoughts of a person. It is achieved by simultaneously combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions. It can be used by a person who has difficulties in speaking or by a person who can hear but could not speak and by normal people to communicate with hearing disabled people. Therefore developing an intelligent system to recognize the sign language and convert it to voice will be massive aid for deaf-mute people to interact with the social world without any hassle and also since the machine takes care of the conversion the need for the normal people to have a good amount of knowledge will be eliminated.
5	Business Model (Revenue Model)	The subscription model is an increasingly popular business model that involves a company selling a service via a subscription as opposed to a one-off product. If the service is going to be helpfull as well as usefull most customers will choose to auto-renew their subscriptions, which

		ensures that they always have the service provided to them. Because of recurring sales, this model allows you to predict what the yearly revenues are going to be, which is fantastic when trying to bring in new investors.
6	Scalability of the Solution	As the technology advances the AI can improvise and more no of people can be incorporated with the system and the system can be extended to incorporate the knowledge of facial expressions and body language too so that there is a complete understanding of the context and tone of the input speech.

3.4 Problem Solution fit

In this activity you are expected to prepare problem - solution fit document and submit for review.



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Data collection	Collection of images and spliting them into train data and test data folders to build the machine learning model.
FR-2	Image processing Tensorflow	TensorFlow is an open-sourced end-to-end platform, a library for multiple machine learning tasks and using this we process the image, image pre-processing includes zooming, shearing, flipping to increase the robustness of the model after it is built.
FR-3	Model building , training and testing	Keras is a high-level neural network library that runs on top of TensorFlow and is used for image pre-processing through imageDataGenerator module and creating Convolutional Neural Network for model building and training.
FR-4	App building	Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. is used to create an web application which will act as an interface between the users and our machine learning model.
FR-5	Storing data collected in IBM cloud	Selecting an IBM Cloud Object Storage account. Selecting the specific geographic region where you want the image template stored. Selecting the IBM Cloud Object Storage bucket where image template is to be stored.
FR-6	Training the model on IBM WATSON STUDIO	IBM Watson Studio is used to build, run, manage and deploy AI models, and optimize decisions anywhere on IBM Cloud Pak for Data. Unite teams, automate AI lifecycles and speed time to value on an open Multicloud architecture.
FR-7	Text to speech	IBM Watson Text to Speech is an API cloud service that converts the text into natural-sounding audio in a variety of languages and voices within an existing application or within Watson Assistant.
FR-8	Integrating the model with the web app and deploying it in IBM cloud	IBM cloud service : To deploy our model on cloud so that it can be created as a service and integrated with the web application created using flask framework.
FR-9	User registration and authentication	Users can register via their gmail account or mobile number and will be authenticated using an OTP and also they will able to set passwords and reset passwords.
FR-10	User login	The registered users can now login using their gmail/mobile number, their password and they are good to go.

4.2 Non-functional Requirements:

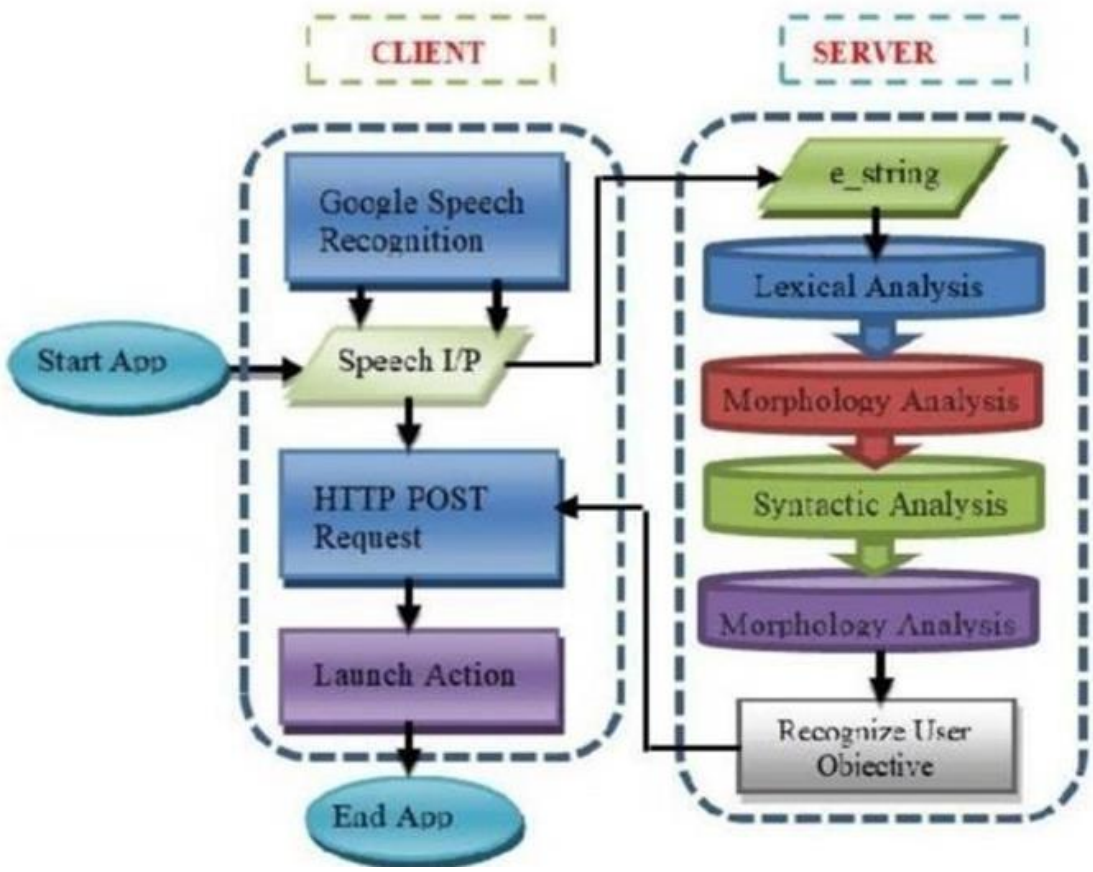
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This web app connected to the AI model service trained on IBM Watson studio AI model service will be usable by any person who wants to communicate with person using sign language but doesn't understand the sign language the app will convert the sign language into voice in the language understandable by the user.
NFR-2	**Security**	Since the model is built on top of the IBM cloud services scalability can also be done with the support of the IBM cloud

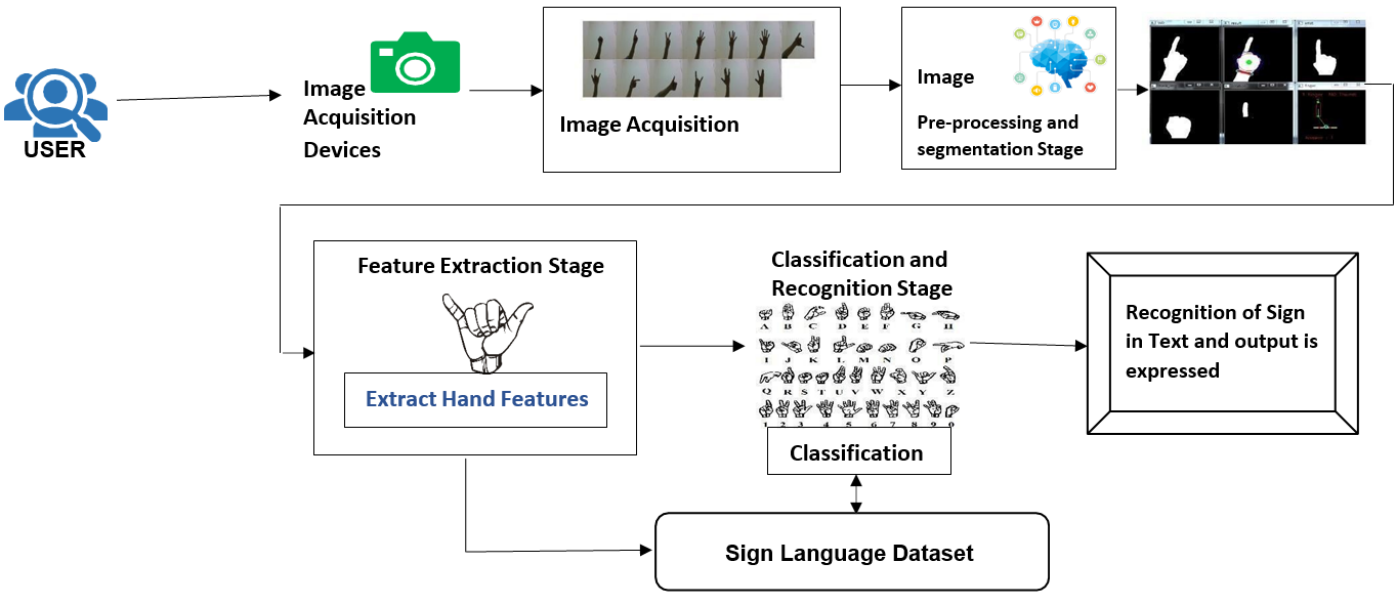
		services in an efficient way.
NFR-3	**Reliability**	The probability of failure-free operation of this service for a specified period in a specified environment will be high and will ensure that the service meets all the requirements to do so. Reliability is a customer-oriented view of software quality and the way this model will be built and tested with care so that it always provides a high quality interactive experience to our customers and clients
NFR-4	**Performance**	An effective model training process and several testing process will be carried out to make accurate predictions of the sign language to satisfy the user requirements.
NFR-5	**Availability**	This service will be available any time and regular updates based on the users feedback an.
NFR-6	**Scalability**	Since the model is built on top of the IBM cloud services scalability can also be done with the support of the IBM cloud services in an efficient way.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2. Solution & Technical Architecture



5.3 User Stories

Our Idea is that every person deserves a fair option to take part in life but life is not fair and it is brutal in certain aspects and it is the responsibility of the fortuned people to help each other and this is the only way to sustain our humanity and to build a functioning good community. The project is souley developed under the motivation to help the people who are believed to be struggling and less fortuned to make their life efficient and smooth.

We really think this project will help and make a huge impact in people life in a positive way and will develop an positive influence and builds confidence to reach out and explore. We have worked on various projects similar to this to help peoples who are less fortuned in life.

Our main motivation was a asian developer ZHANG his ideation is ZHANG:

his team hope DeepASL can help people who are deaf and hard of hearing by serving as a real-time translator. It could be especially useful in emergency situations, Zhang says, when waiting for a translator could cost precious minutes. The device, which could be integrated with a phone, tablet or computer, can also help teach ASL, Zhang says. Since more than 90 percent of deaf children are born to parents who are hearing, there is a large community of adults who need to learn ASL quickly. DeepASL acts as a digital tutor and gives learners feedback on whether they are signing correctly.

Zhang has applied for a patent and hopes to bring the device to market within a year. Built on affordable technology (Leap Motion's motion capture system retails for $78), it may become more widely available than previous efforts``

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

IDEATION PHASE

TITLE	DESCRIPTION	DATE
Literature Survey	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	**14 OCTOBER 2022**
Empathy Map for Web Phishing Detection	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	**7 OCTOBER 2022**
Problem Statement	Prepare the problem statement document	**18 OCTOBER 2022**
Brainstorming Idea Generation Prioritization	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	**1 NOVEMBER 2022**

PROJECT DESIGN PHASE-I

TITLE	DESCRIPTION	DATE
Problem Solution Fit	Prepare problem - solution fit document.	**14 OCTOBER 2022**
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	**14 OCTOBER 2022**
Solution Architecture	Prepare solution architecture document.	14 OCTOBER 2022

PROJECT DESIGN PHASE-II

TITLE	DESCRIPTION	DATE
Solution Requirements	Prepare the functional requirement document.	**15 OCTOBER 2022**
Customer Journey Map	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	**1 NOVEMBER 2022**
Data Flow Diagrams and User Stories	Draw the data flow diagrams and submit for review.	**1 NOVEMBER 2022**
Technology Stack	Prepare the technology architecture diagram	**15 OCTOBER 2022**

PROJECT PLANNING PHASE

TITLE	DESCRIPTION	DATE
Project Planning	Prepare the planning for this project	**01 NOVEMBER 2022**
Milestone and Activity List	Prepare the milestones & activity list of the project	**01 NOVEMBER 2022**

PROJECT DEVELOPMENT

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Nov 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	7 Nov 2022	12 Nov 2022	20	17 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	Medium	Yashwanth, Raj Kumar, Vettri Chezhian, Naveen Kumar.

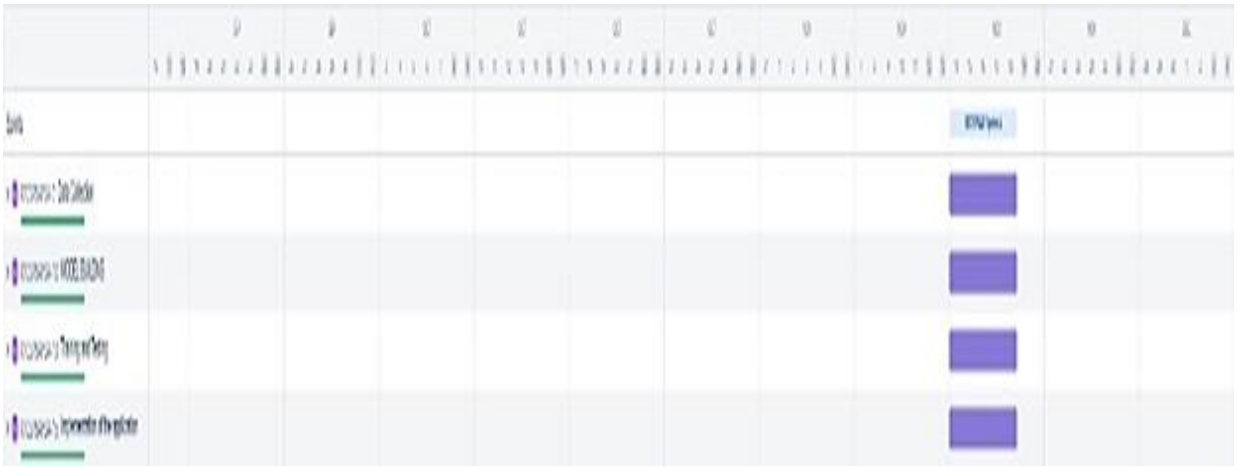
Sprint-1	Login	USN-1	As a user, I can log into the application by enteringemail & password	1	High	Yashwanth, Raj Kumar, Vettri Chezhian, Naveen Kumar.
Sprint-1	Data Collection	USN-3	Gathering the information from various resources	1	Medium	Yashwanth, Raj Kumar, Vettri Chezhian, Naveen Kumar.
Sprint-1	Data Preprocessing	USN-4	To Convert and clean the raw data	2	High	Yashwanth, Raj Kumar, Vettri Chezhian, Naveen Kumar.
Sprint-2	Model Building	USN-5	Using cleaned dataset, Model can be build usingML Algorithm	2	High	Yashwanth, Raj Kumar, Vettri Chezhian, Naveen Kumar.
Sprint-2		USN-6	Training the classification model	1	High	Yashwanth, Raj Kumar, Vettri Chezhian, Naveen Kumar.
Sprint-3	Application Building	USN-7	Building Python code and run the application	1	Medium	Yashwanth, Raj Kumar, Vettri Chezhian, Naveen Kumar.
Sprint-3		USN-8	Predicted Result	1	Medium	Yashwanth, Raj Kumar, Vettri Chezhian, Naveen Kumar.
Sprint-4		USN-9	Deployed on IBM Cloud	2	High	Yashwanth, Raj Kumar, Vettri Chezhian, Naveen Kumar.

6.3 Reports from JIRA

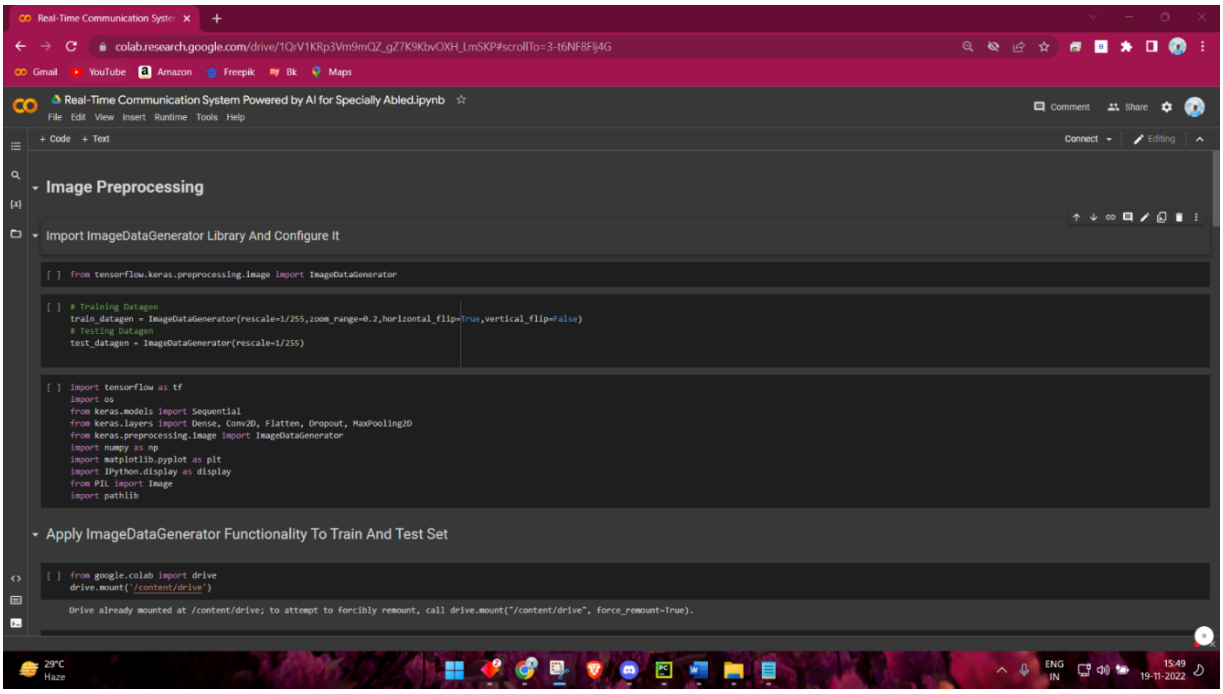
Jira helps teams plan, assign, track, report, and manage work and brings teams together foreverything from agile software development and customer support to start-ups and enterprises.

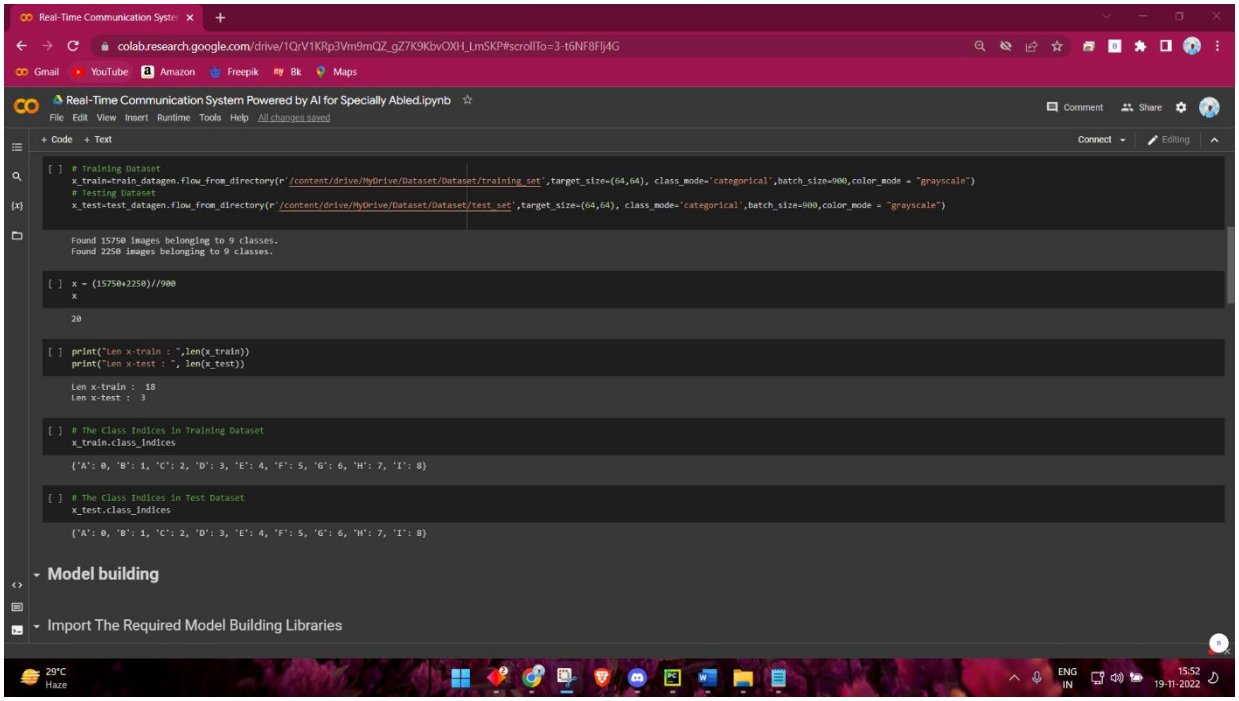
Software teams build better with Jira Software, the #1 tool for agile teams. As a Jira administrator,

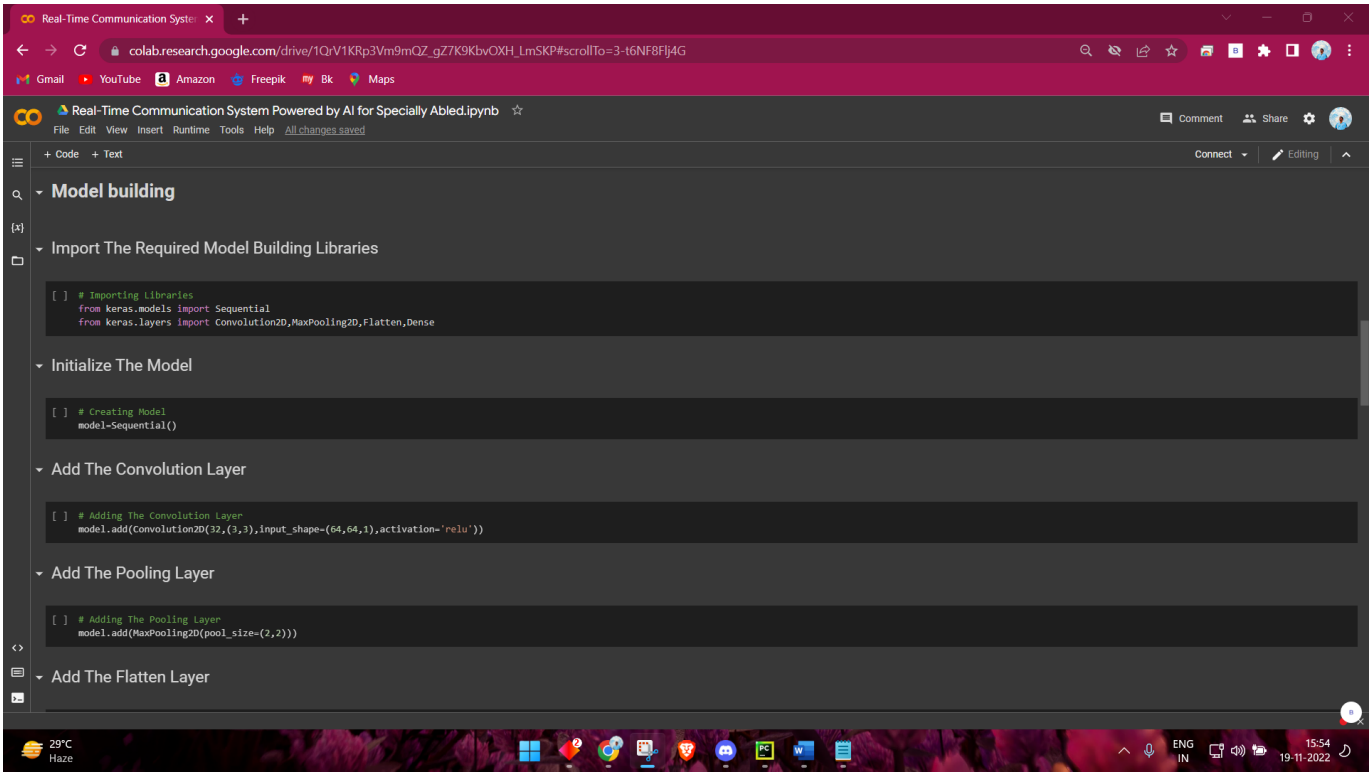
you can create project categories so your team can view work across related projects in one place.Your team can use categories in advanced search, filters, reports, and more.

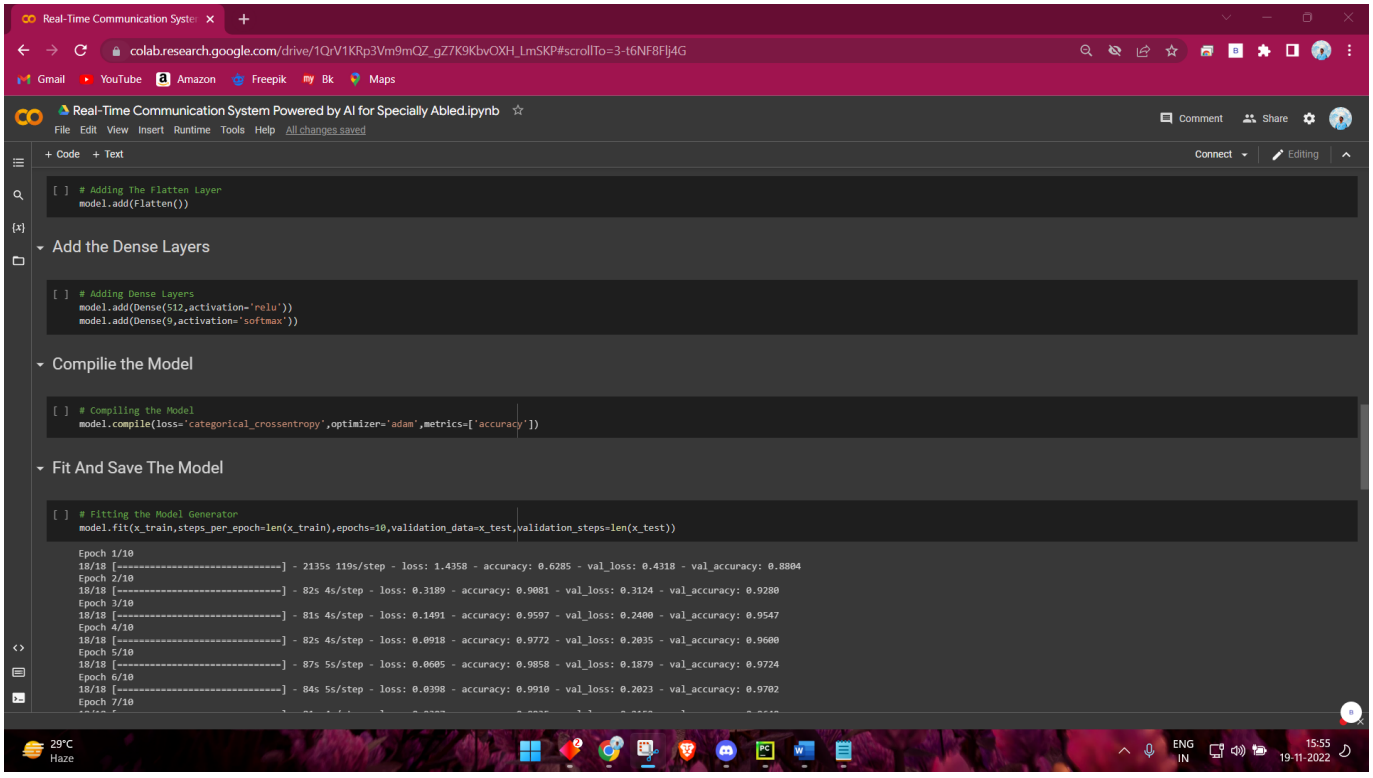


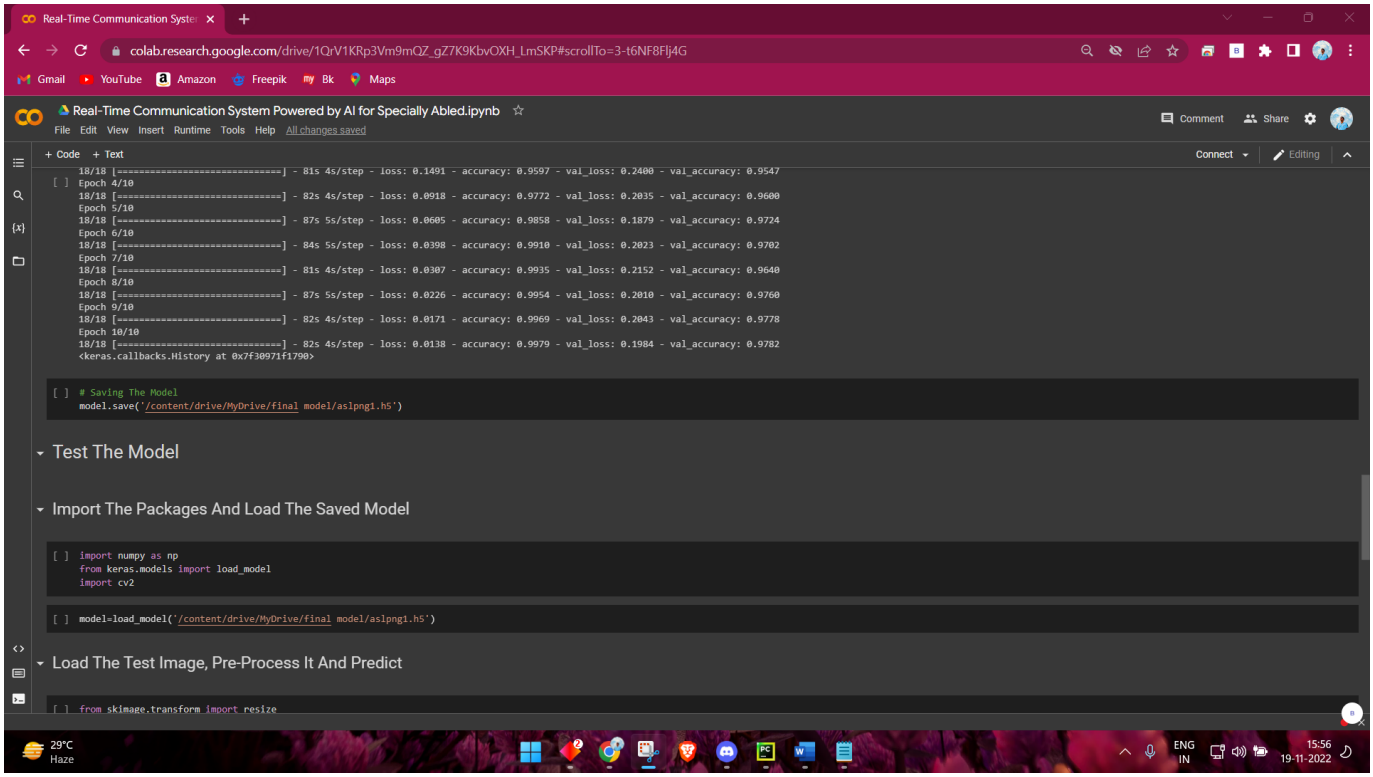
7. CODING & SOLUTIONING

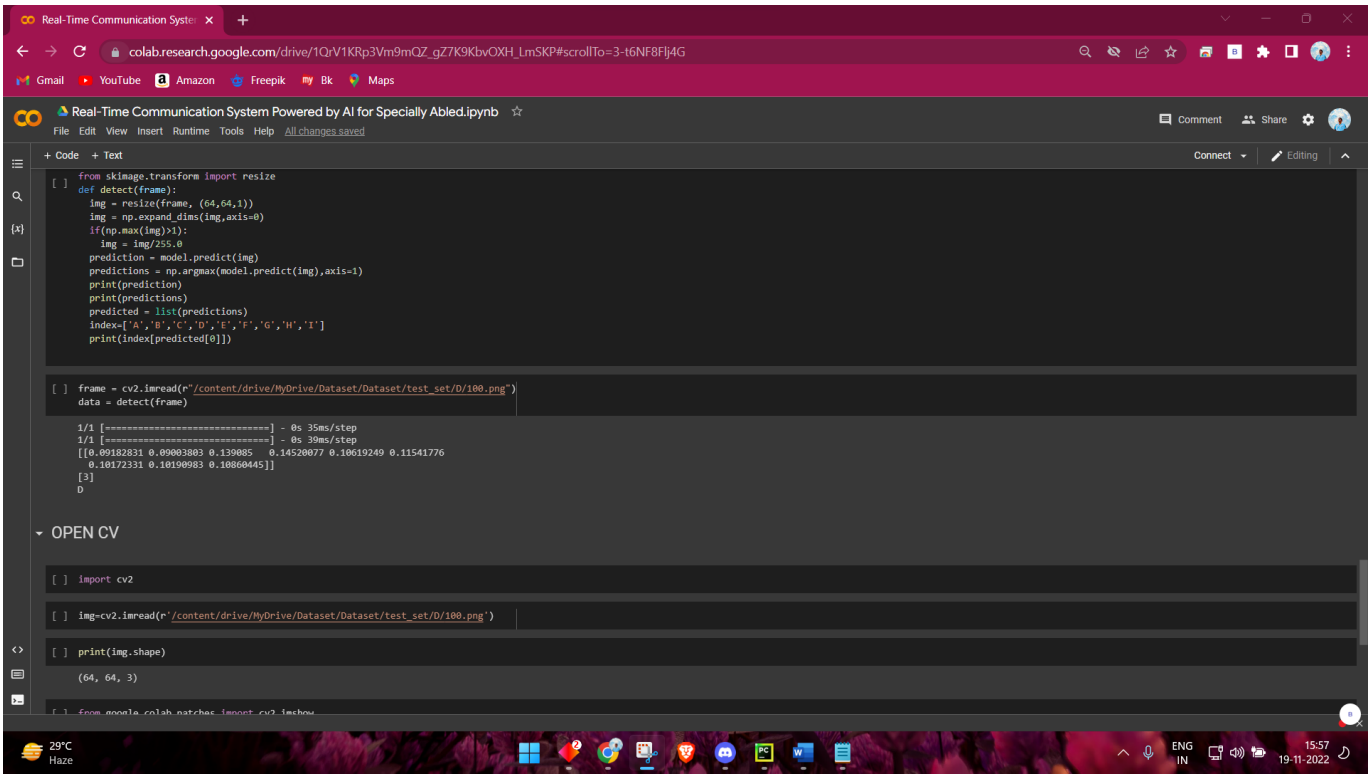


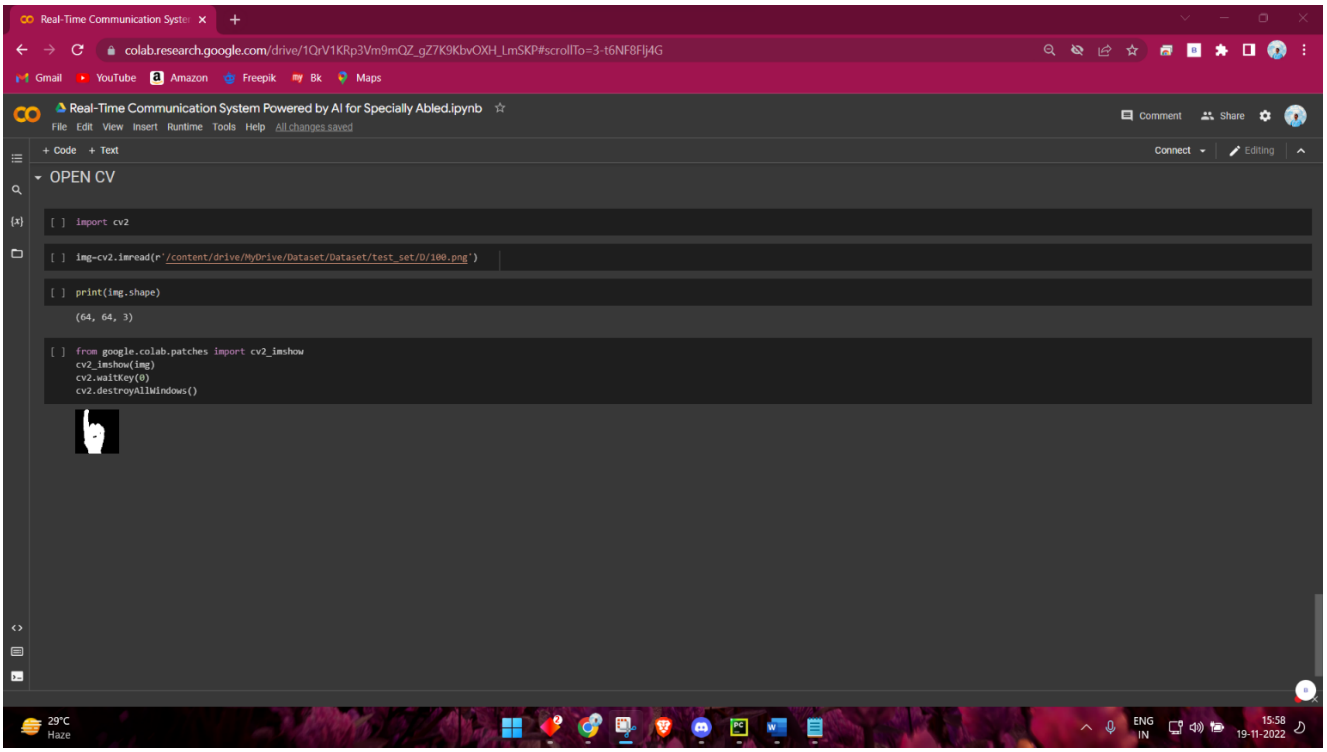



Real-Time Communication Syste
colab.research.google.com/drive/1QrV1KRp3Vm9mQZ_gZ7K9KbvOXH_LmSKP#scrollTo=3-t6NF8Flj4G
Gmail YouTube Amazon Freepik Bk Maps
Real-Time Communication System Powered by AI for Specially Abled.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Connect Editing
Model building
Import The Required Model Building Libraries
[ ] # Importing Libraries
from keras.models import Sequential
from keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
Initialize The Model
[ ] # Creating Model
model=Sequential()
Add The Convolution Layer
[ ] # Adding The Convolution Layer
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
Add The Pooling Layer
[ ] # Adding The Pooling Layer
model.add(MaxPooling2D(pool_size=(2,2)))
Add The Flatten Layer
29°C Haze
ENG IN
15:54 19-11-2022

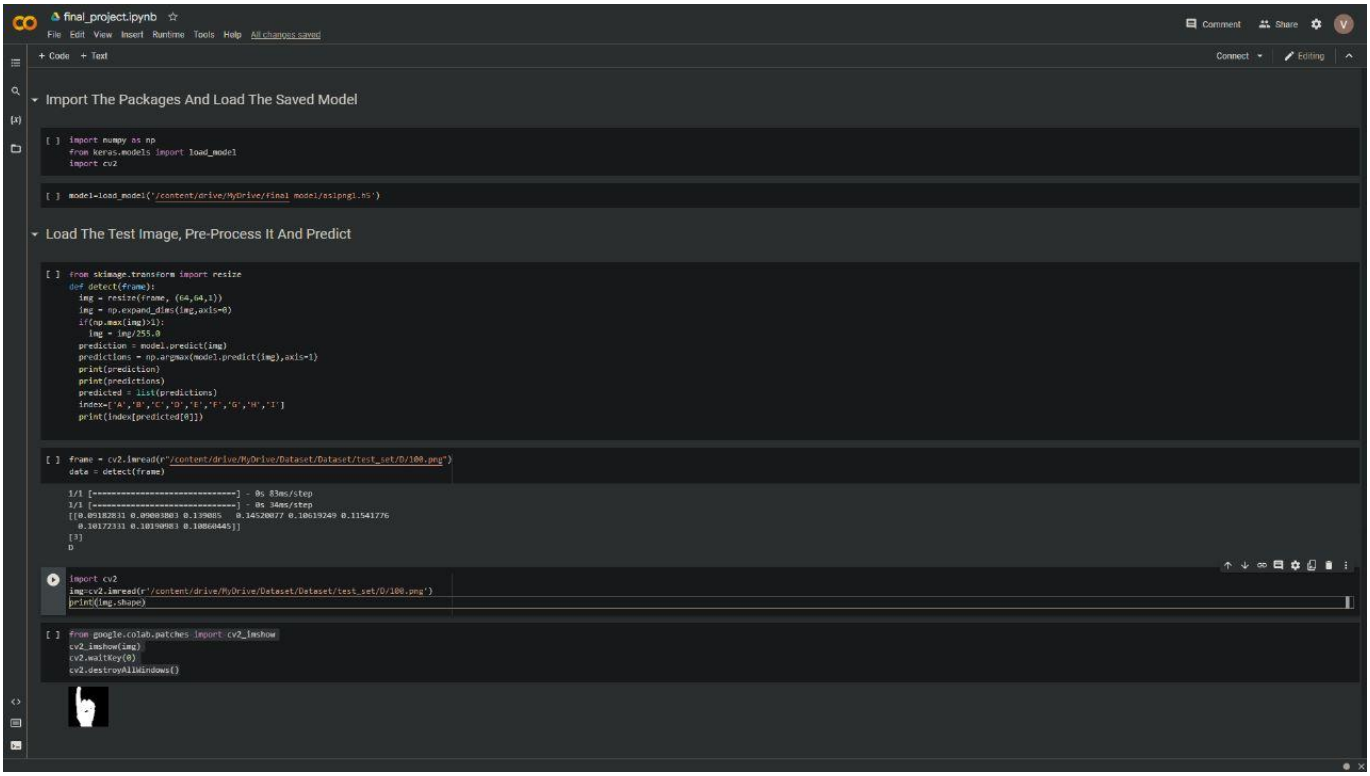

Real-Time Communication Syste
colab.research.google.com/drive/1QrV1KRp3Vm9mQZ_gZ7K9KbvOXH_LmSKP#scrollTo=3-t6NF8Flj4G
Gmail YouTube Amazon Freepik Bk Maps
Real-Time Communication System Powered by AI for Specially Abled.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Connect Editing
[ ] # Adding The Flatten Layer
model.add(Flatten())
Add the Dense Layers
[ ] # Adding Dense Layers
model.add(Dense(512,activation='relu'))
model.add(Dense(9,activation='softmax'))
Compilie the Model
[ ] # Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
Fit And Save The Model
[ ] # Fitting the Model Generator
model.fit(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
Epoch 1/10
18/18 [==============================] - 2135s 119s/step - loss: 1.4358 - accuracy: 0.6285 - val_loss: 0.4318 - val_accuracy: 0.8804
Epoch 2/10
18/18 [==============================] - 82s 4s/step - loss: 0.3189 - accuracy: 0.9081 - val_loss: 0.3124 - val_accuracy: 0.9280
Epoch 3/10
18/18 [==============================] - 81s 4s/step - loss: 0.1491 - accuracy: 0.9597 - val_loss: 0.2400 - val_accuracy: 0.9547
Epoch 4/10
18/18 [==============================] - 82s 4s/step - loss: 0.0918 - accuracy: 0.9772 - val_loss: 0.2035 - val_accuracy: 0.9600
Epoch 5/10
18/18 [==============================] - 87s 5s/step - loss: 0.0605 - accuracy: 0.9858 - val_loss: 0.1879 - val_accuracy: 0.9724
Epoch 6/10
18/18 [==============================] - 84s 5s/step - loss: 0.0398 - accuracy: 0.9910 - val_loss: 0.2023 - val_accuracy: 0.9702
Epoch 7/10
29°C Haze
ENG IN
15:55 19-11-2022

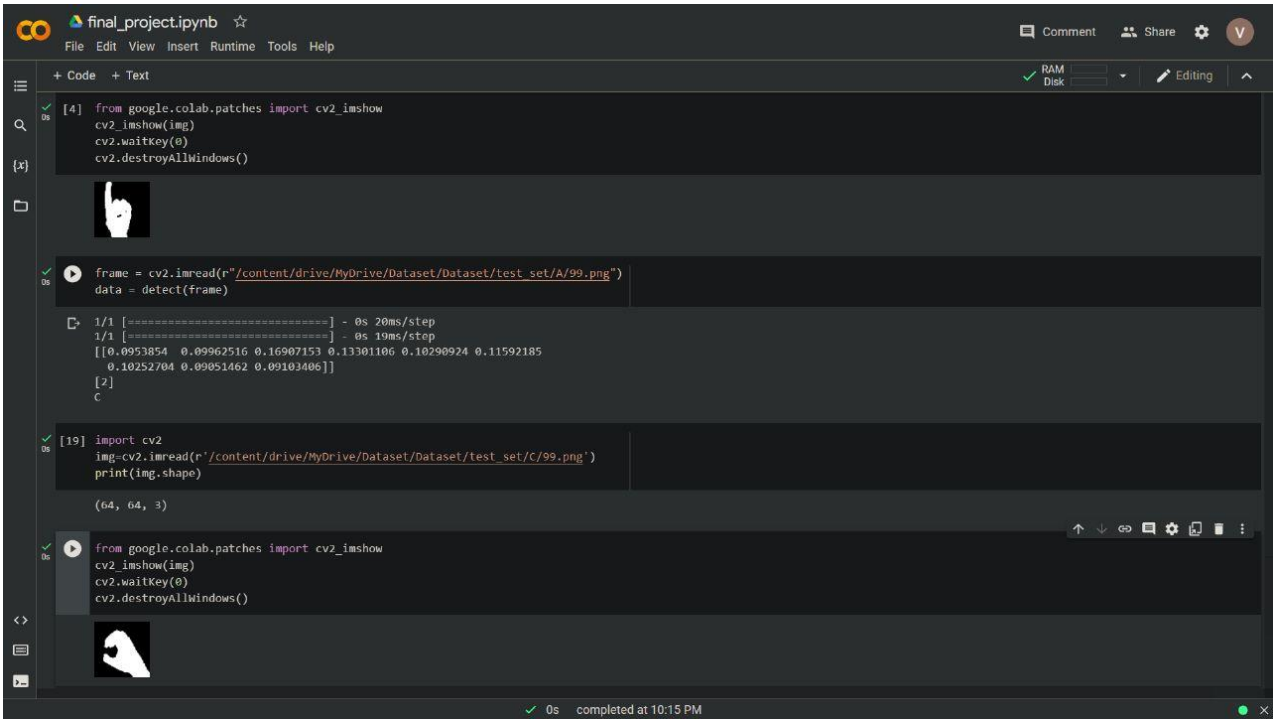

Real-Time Communication Syste
colab.research.google.com/drive/1QrV1KRp3Vm9mQZ_gZ7K9KbvOXH_LmSKP#scrollTo=3-t6NF8Flj4G
Gmail YouTube Amazon Freepik Bk Maps
Real-Time Communication System Powered by AI for Specially Abled.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Connect Editing
18/18 [==========] - 81s 4s/step - loss: 0.1491 - accuracy: 0.9597 - val_loss: 0.2400 - val_accuracy: 0.9547
Epoch 4/10
18/18 [==========] - 82s 4s/step - loss: 0.0918 - accuracy: 0.9772 - val_loss: 0.2035 - val_accuracy: 0.9600
Epoch 5/10
18/18 [==========] - 87s 5s/step - loss: 0.0605 - accuracy: 0.9858 - val_loss: 0.1879 - val_accuracy: 0.9724
Epoch 6/10
18/18 [==========] - 84s 5s/step - loss: 0.0398 - accuracy: 0.9910 - val_loss: 0.2023 - val_accuracy: 0.9702
Epoch 7/10
18/18 [==========] - 81s 4s/step - loss: 0.0307 - accuracy: 0.9935 - val_loss: 0.2152 - val_accuracy: 0.9640
Epoch 8/10
18/18 [==========] - 87s 5s/step - loss: 0.0226 - accuracy: 0.9954 - val_loss: 0.2010 - val_accuracy: 0.9760
Epoch 9/10
18/18 [==========] - 82s 4s/step - loss: 0.0171 - accuracy: 0.9969 - val_loss: 0.2043 - val_accuracy: 0.9778
Epoch 10/10
18/18 [==========] - 82s 4s/step - loss: 0.0138 - accuracy: 0.9979 - val_loss: 0.1984 - val_accuracy: 0.9782
<keras.callbacks.History at 0x7f30971f1790>
# Saving The Model
model.save('/content/drive/MyDrive/final model/aslpng1.h5')
Test The Model
Import The Packages And Load The Saved Model
import numpy as np
from keras.models import load_model
import cv2
model=load_model('/content/drive/MyDrive/final model/aslpng1.h5')
Load The Test Image, Pre-Process It And Predict
from skimage.transform import resize
29°C Haze
ENG IN
15:56 19-11-2022


Real-Time Communication Syste
colab.research.google.com/drive/1QrV1KRp3Vm9mQZ_gZ7K9KbvOXH_LmSKP#scrollTo=3-t6NF8Flj4G
Gmail YouTube Amazon Freepik Bk Maps
Real-Time Communication System Powered by AI for Specially Abled.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Connect Editing
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)
predictions = np.argmax(model.predict(img),axis=1)
print(prediction)
print(predictions)
predicted = list(predictions)
index=['A','B','C','D','E','F','G','H','I']
print(index[predicted[0]])
frame = cv2.imread(r"/content/drive/MyDrive/Dataset/Dataset/test_set/D/100.png")
data = detect(frame)
1/1 [==========] - 0s 35ms/step
1/1 [==========] - 0s 39ms/step
[[0.09182831 0.09003803 0.139085 0.14520077 0.10619249 0.11541776
0.10172331 0.10190983 0.10860445]]
[3]
D
OPEN CV
import cv2
img=cv2.imread(r'/content/drive/MyDrive/Dataset/Dataset/test_set/D/100.png')
print(img.shape)
(64, 64, 3)
29°C Haze
ENG IN
15:57 19-11-2022


Real-Time Communication Syste
colab.research.google.com/drive/1QrV1KRp3Vm9mQZ_gZ7K9KbvOXH_LmSKP#scrollTo=3-t6NF8FJj4G
Gmail YouTube Amazon Freepik Bk Maps
Real-Time Communication System Powered by AI for Specially Abled.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Connect Editing
OPEN CV
[ ] import cv2
[ ] img=cv2.imread(r'/content/drive/MyDrive/Dataset/Dataset/test_set/D/100.png')
[ ] print(img.shape)
(64, 64, 3)
[ ] from google.colab.patches import cv2_imshow
cv2_imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()
29°C
Haze
ENG
IN
15:58
19-11-2022

8. TESTING

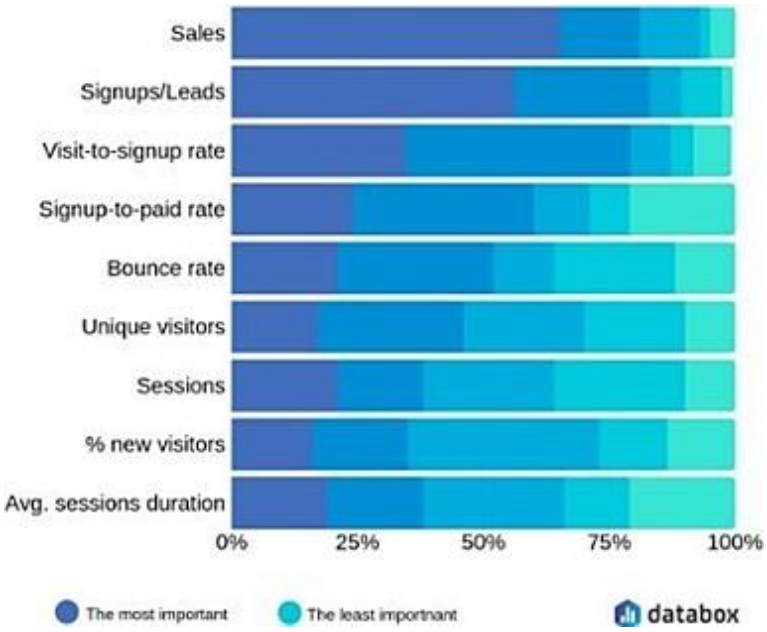

final_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Connect Editing
Import The Packages And Load The Saved Model
[ ] import numpy as np
from keras.models import load_model
import cv2
[ ] model=load_model('/content/drive/MyDrive/Final model/aslpng1.h5')
Load The Test Image, Pre-Process It And Predict
[ ] 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)
predictions = np.argmax(model.predict(img),axis=1)
print(prediction)
print(predictions)
predicted = list(predictions)
index=['A','B','C','D','E','F','G','H','I']
print(index[predicted[0]])
[ ] frame = cv2.imread(r"/content/drive/MyDrive/Dataset/Dataset/test_set/D/100.png")
data = detect(frame)
1/1 [==============================] - 0s 83ms/step
1/1 [==============================] - 0s 34ms/step
[[0.09182831 0.09083803 0.139085 0.14520077 0.10619249 0.11541776
0.10172331 0.10190983 0.10860445]]
[3]
D
import cv2
img=cv2.imread(r'/content/drive/MyDrive/Dataset/Dataset/test_set/D/100.png')
print(img.shape)
[ ] from google.colab.patches import cv2_imshow
cv2_imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()



9. RESULTS

9.1 Performance Metrics

- ❖ The proposed procedure was implemented and tested on a set of images.
- ❖ The training database consists of 15750 images of Alphabets from "A" to "I", while the testing database consists of 2250 images of Alphabets from "A" to "I".
- ❖ Once the gesture is recognized 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 kind of 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 recognises them and displays the equivalent Alphabet on the screen. Deaf-mute 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.

13. APPENDIX

Source Code:

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()
```

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="static/Banner-Heading-Image.css">
    <link rel="stylesheet" href="staticNavbar-Centered-Brand.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">Student of PANIMALAR ENGINEERING
COLLEGE.<br><br>1. <strong>Vettri chezhian P</strong> 211419205175<br>2.
                    <strong>Raj kumar K</strong> 211419205136<br>3.
<strong>Yashwanth M</strong> 211419205186<br>4.
                    <strong>Naveen kumar M</strong> 211419205115
                  </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"><img src="static/ASL_Alphabet.jpg"
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>
```

Navbar-Centered-Brand.css

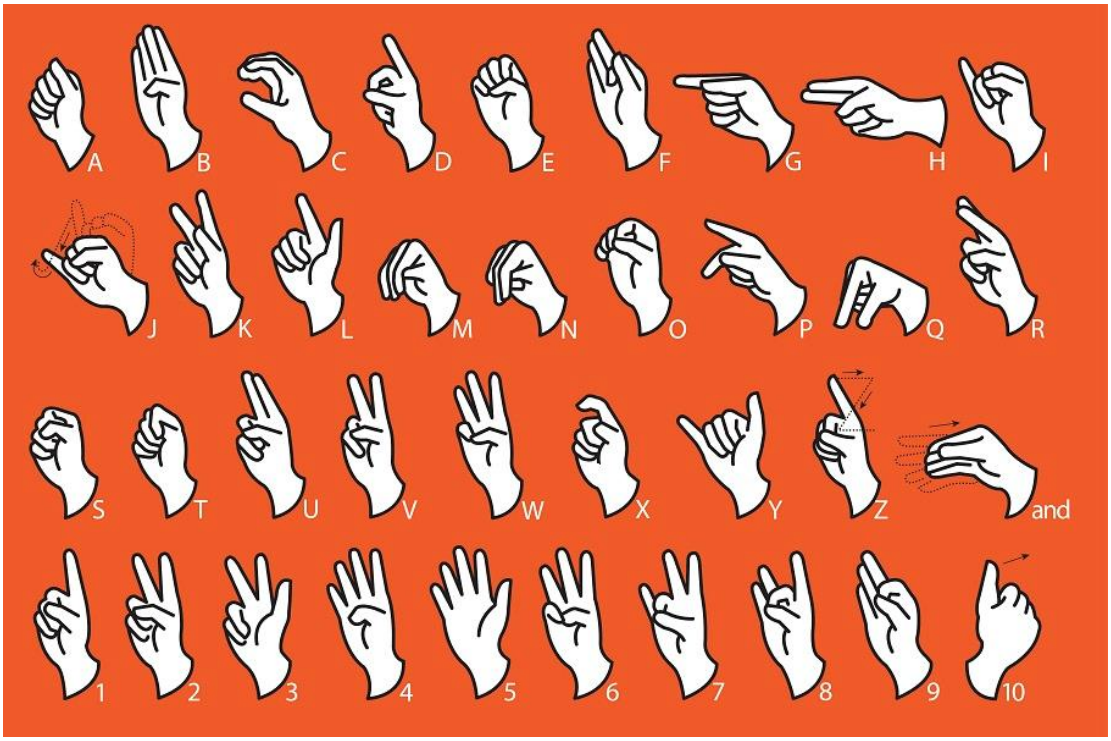
```
.bs-icon {
  --bs-icon-size: .75rem;
  display: flex;
  flex-shrink: 0;
  justify-content: center;
  align-items: center;
  font-size: var(--bs-icon-size);
  width: calc(var(--bs-icon-size) * 2);
  height: calc(var(--bs-icon-size) * 2);
  color: var(--bs-primary);
}

.bs-icon-xs {
  --bs-icon-size: 1rem;
  width: calc(var(--bs-icon-size) * 1.5);
  height: calc(var(--bs-icon-size) * 1.5);
}

.bs-icon-sm {
  --bs-icon-size: 1rem;
}

.bs-icon-md {
  --bs-icon-size: 1.5rem;
}

.bs-icon-lg {
  --bs-icon-size: 2rem;
}

.bs-icon-xl {
  --bs-icon-size: 2.5rem;
}

.bs-icon.bs-icon-primary {
  color: var(--bs-white);
  background: var(--bs-primary);
```

```
}

.bs-icon.bs-icon-primary-light {
  color: var(--bs-primary);
  background: rgba(var(--bs-primary-rgb), .2);
}

.bs-icon.bs-icon-semi-white {
  color: var(--bs-primary);
  background: rgba(255, 255, 255, .5);
}

.bs-icon.bs-icon-rounded {
  border-radius: .5rem;
}

.bs-icon.bs-icon-circle {
  border-radius: 50%;
}
```

banner-Heading-Image.css

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

ASL_Alphabet.jpg



camera.py

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


class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        self.model = load_model('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 = load_img('image.jpg', target_size=(64, 64))
        x = img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame, 'The Predicted Alphabet is: ' + str(self.index[self.y]), (100, 50),
                                    cv2.FONT_HERSHEY_SIMPLEX,
                    1, (0, 0, 0), 3)
        ret, jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

GitHub

https://github.com/IBM-EPBL/IBM-Project-5256-1658752692

Project Demo Link

https://ibmprojecttnt2022tmid01421.s3.jp-tok.cloud-object-storage.appdomain.cloud/PROJECT%20DEMO%20VIDEO.mp4