# IBM – NALAIYA THIRAN

# PROJECT REPORT

# On

# REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED


**Submitted by**

| | |
|---|---|
| **SUJITH KUMAR R** | **(AC19UCS118)** |
| **SOUNDAR SRIRAM J** | **(AC19UCS111)** |
| **SRIDHAR C** | **(AC19UCS113)** |
| **SRINATH M** | **(AC19UCS155)** |

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**ADHIYAMAAN COLLEGE OF ENGINEERING**

DR. M.G.R NAGAR, HOSUR-635130



**REPO ID: IBM-Project-12414-1659450858**

**TEAM ID: PNT2022TMID08035**



**FACULTY MENTOR: Dr. N. MORATANCH**

**INDUSTRIAL MENTOR: Prof. DIVYA**

# CONTENTS

# CHAPTER 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

The project's purpose is to create a system that translates sign language into a human understandable language so that ordinary people may understand it.

# CHAPTER 2

# LITERATURE SURVEY

## 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


- Sign Language Recognition System for People with Disability:

https://drive.google.com/file/d/1-iCgw_ai-9N111UbflsEUPxkY3OFzMWY/view?usp=drivesdk

- A Real Time System for Two Ways Communication of Hearing and Speech Impaired People:

https://drive.google.com/file/d/199cSfjzLRQwfU5chMNbCQMqcGZTeC9eD/view?usp=sharing

- Artificial Intelligence Enabled virtual sixth sense application for the disabled:

https://drive.google.com/file/d/1NZ0MMd4NqhPUeQAJ6X_R256bFL4_Bo2h/view?usp=drivesdk

- Based Hand Sign Detection System For Deaf-Mute People:

https://drive.google.com/file/d/1-lpUmznb1KnxfN724X8A-jPdeAgj57Fd/view?usp=drivesdk

**2.3 PROBLEM STATEMENT DEFINITION**

**Why do we need a real time communication system for the specially-abled?**

According to the times now survey, the Indian population consists of about 30 percent disabled people, and of that 20 percent are deaf and mute. The only chance of communication is the sign language but it's practically not feasible that everyone studies the sign language. Technology has risen to unprecedented rates which also comes with a leeway for the disabled people. With the help of technology, 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.

**OUR PLAN:**

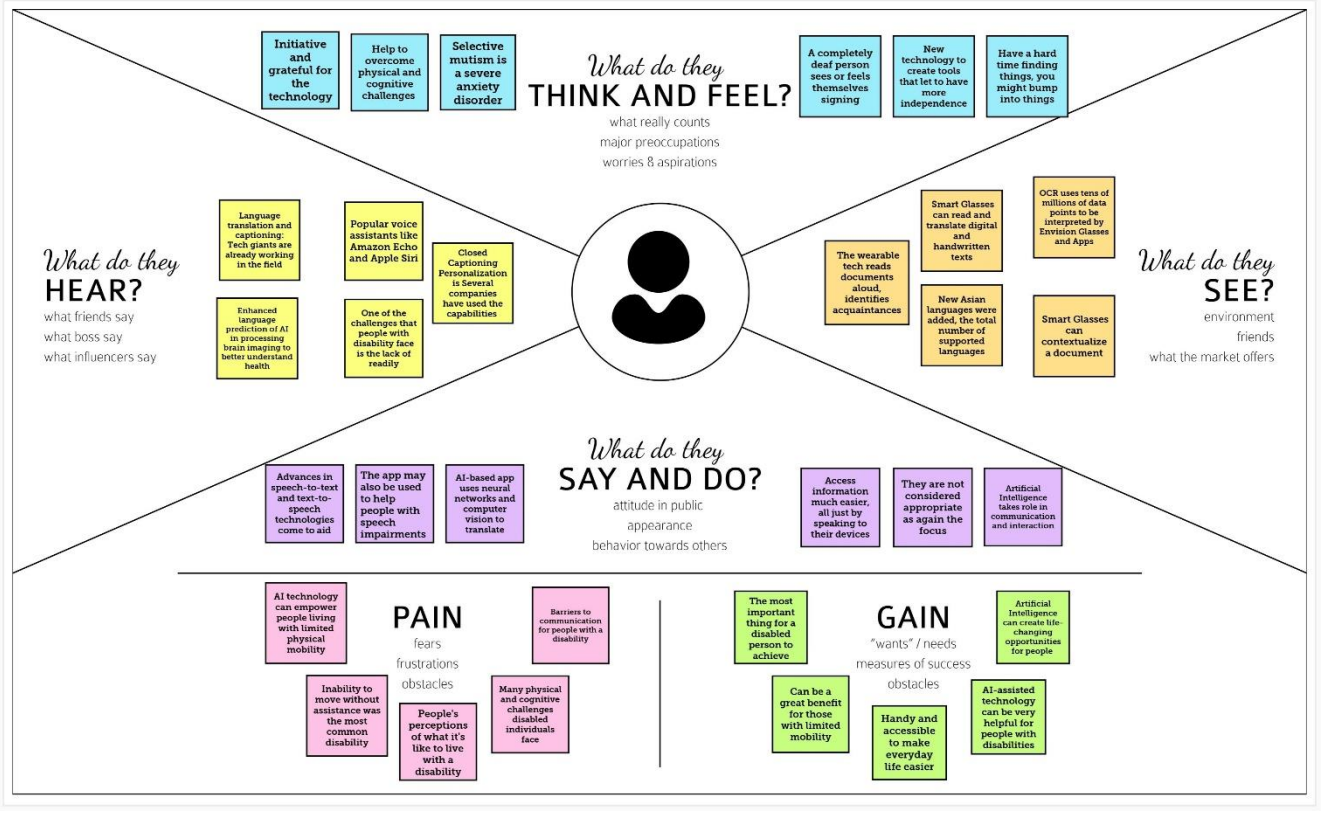
The aim of this project is to create a software that does not only convert sign language into text and speech but also translates speech into sign language in real time and as quick as the person speaks. We will be using a deep learning model like CNN for this project. CNN is used for image classification and classifies the object into the respective classes and does the object detection accordingly. An app is built which uses this model. This app enables deaf and mute people to convey their information using signs which gets converted to human-understandable language and speech is given as output.

# CHAPTER 3

# IDEATION & PROPOSED SOLUTION

## 3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



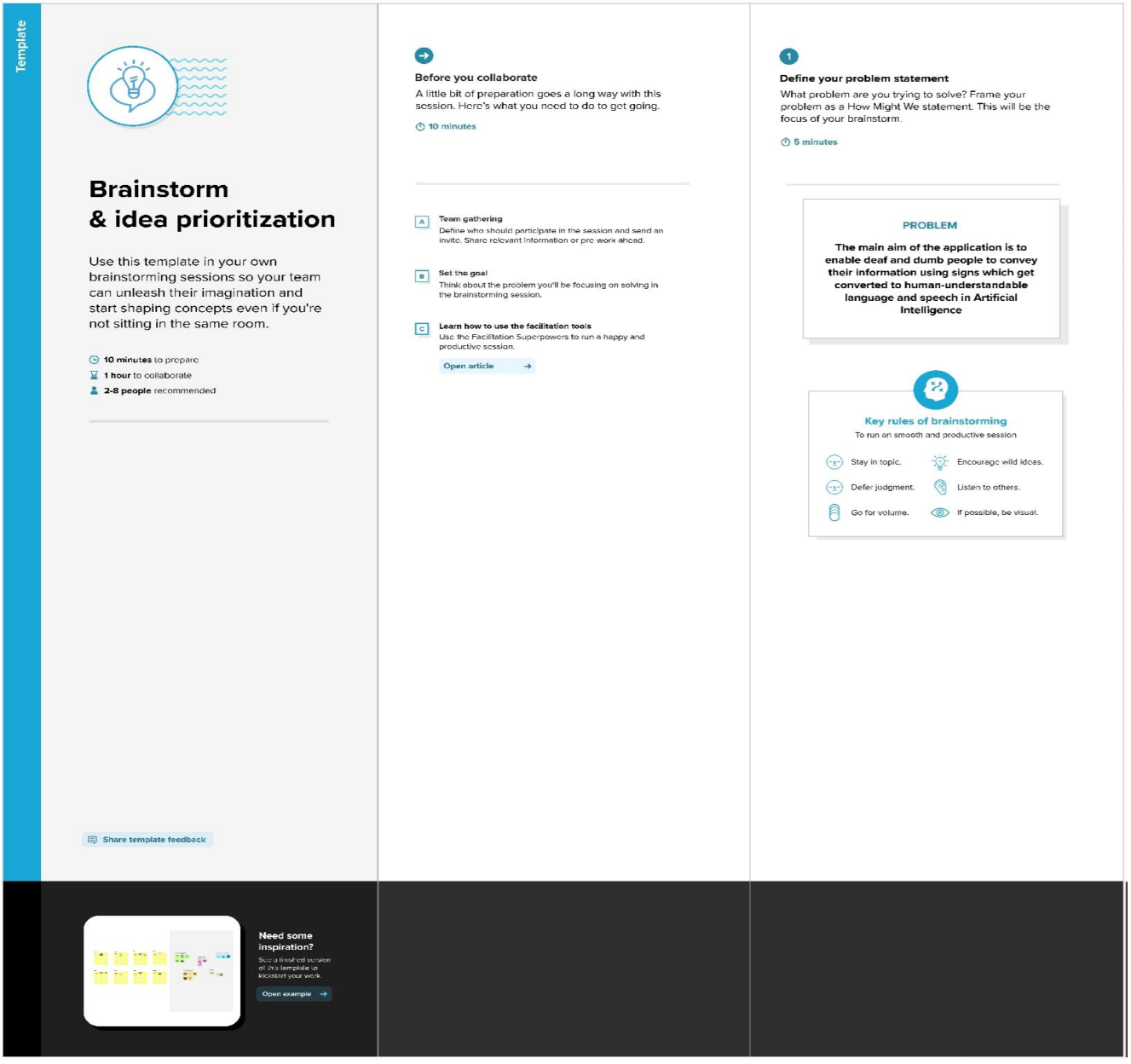

## 3.2 IDEATION & BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in

your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

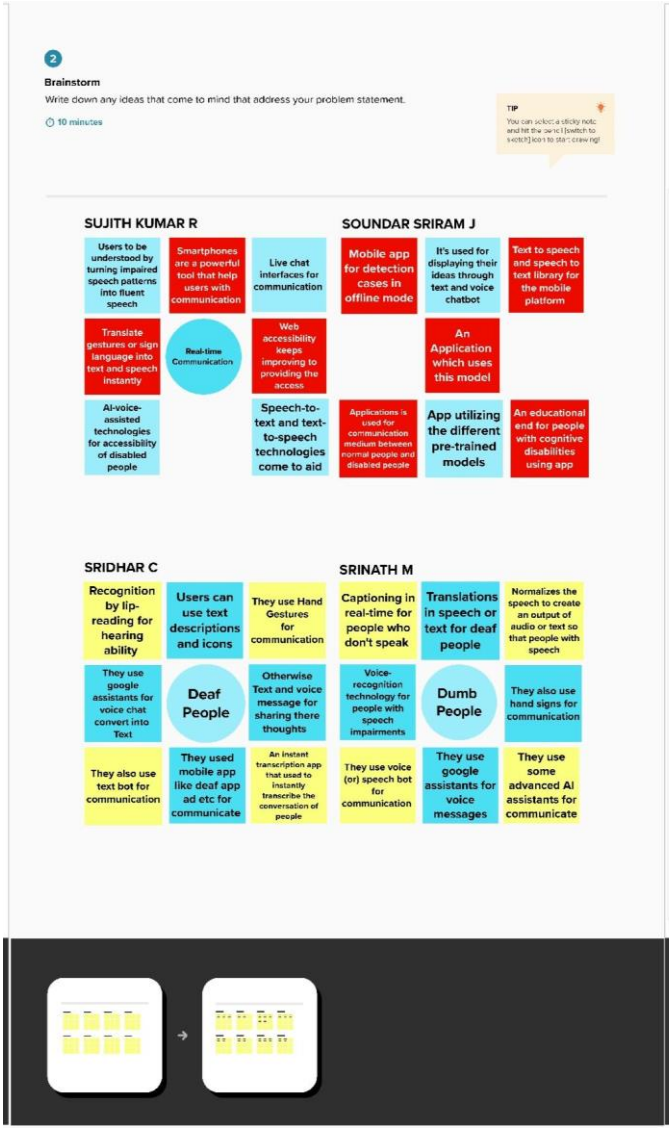
**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

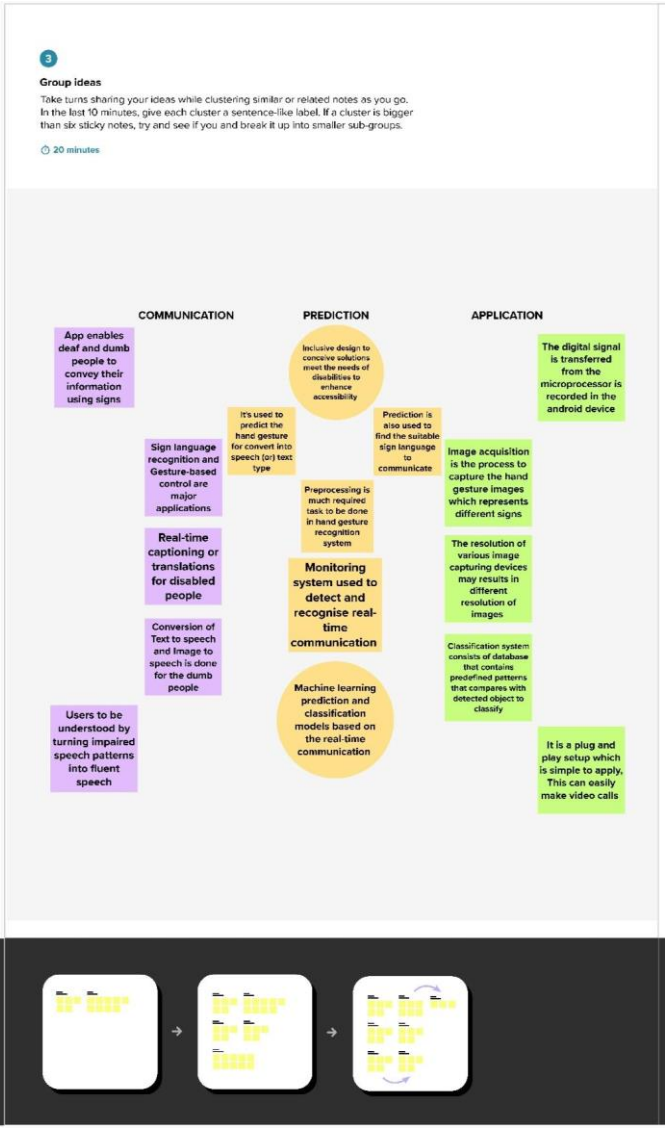
In this step team members gather and provide their ideas and collaborate those ideas and select their problem statement. The ideas should be relevant to their problem statement.



## Step-2: Brainstorm, Idea Listing and Grouping

In this step they put their ideas and views which are prioritized based on their importance and the ideas are grouped. These ideas are categorized according to their relevant classifications.

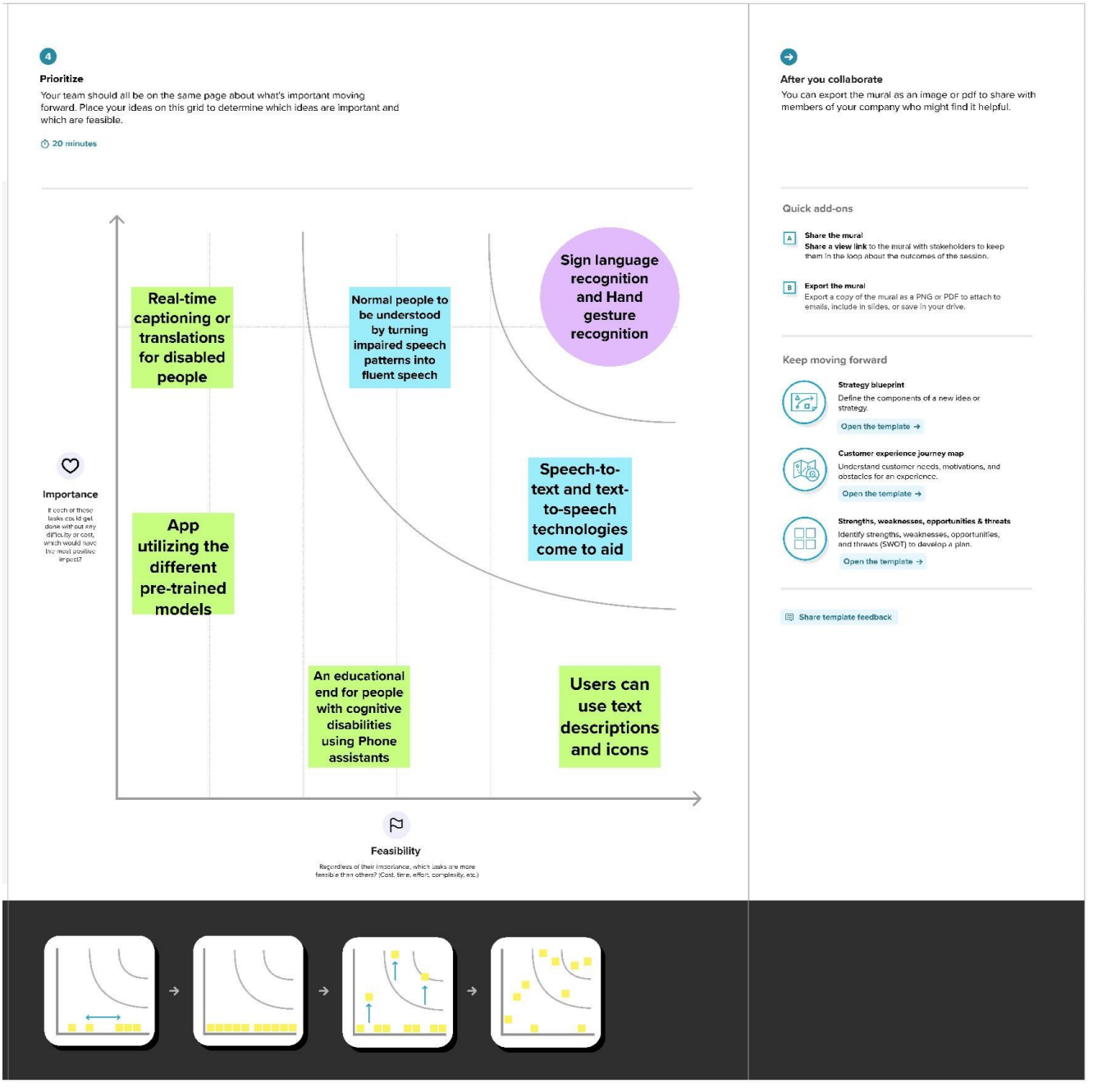



## Step-3: Idea Prioritization

As mentioned, idea prioritization is just a part of the idea management process. Having a structured idea management process and a systematic way of gathering, evaluating and prioritizing new ideas

takes time. To make it work, the entire idea management process should be integrated into everyday ways of working.



## 3.3 PROPOSED SOLUTION

- **Problem Statement:**

An application for deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech in Artificial Intelligence

- **Idea / Solution description:**

By using Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation

- **Novelty / Uniqueness:**

We are using a convolution neural network to create a model that is trained on different hand gestures and an app is built for the use this mode

- **Social Impact / Customer Satisfaction:**

Communicating with others and being connected in the society and remove accessibility barriers

- **Business Model:**

By Using: Better communication with the disabled and Financial

By Without Using: Can’t Communicate and leads to loneliness

- **Scalability of the Solution:**

Enhance people with disabilities to step into a world where their are facing difficulties in communication

# 3.4 PROBLEM SOLUTION FIT

Define CS, fit into CC

**1. CUSTOMER SEGMENT(S)** CS

- Normal People, Who needs to communicate with specially abled.
- Deaf People
- Dumb People

**6. CUSTOMER CONSTRAINTS** CC

Artificial Intelligence technology solutions, discover how accessibility for people can be enhanced.

**5. AVAILABLE SOLUTIONS** AS

- Voice Conversion system with hand gestures recognition and translation will be very useful to have a proper conversation between a normal people and an impaired person in any language.
- An app is built which enables deaf and dumb people to convey their information using signs.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

**2. JOBS-TO-BE-DONE / PROBLEMS** J&P

- Input is given as Hand gestures image, which undergoes image preprocessing and voice recognition.
- In Neural network, the hand gestures are trained by Convolution Neural Network.

**9. PROBLEM ROOT CAUSE** RC

- Communication between deaf-mute and a normal people has always been a challenging task.
- It is very difficult for mute people to convey their message to normal people.
- Normal people are not trained on hand sign language, and in emergency times it is very difficult.

**7. BEHAVIOUR** BE

What do to address the problem

- Artificial Intelligence model that converts sign language into a speech that can be understood by normal people.
- An application built for efficient usage.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

**3. TRIGGERS** TR

- The benefit of the 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.

**4. EMOTIONS: BEFORE / AFTER** EM

BEFORE: Normal people are not aware of hand signs and Communication is difficult between the normal people and specially abled.

AFTER: Communication is good and efficient between the normal people and specially abled and Normal people can learn sign language.

**10. YOUR SOLUTION** SL

- An application for deaf and dumb people to convey their information using signs.
- By using voice conversion system with hand gesture recognition and translation will be very useful to have a proper conversation.

**8.CHANNELS of BEHAVIOR** CH

**8.1 ONLINE**

- A simple and beautiful user interface is used and supports different languages.
- Accurate prediction and used speech to text & text to speech.

**8.2 OFFLINE**

- Communication is made between the normal people and specially abled.
- Normal people used to learn sign language.

# CHAPTER 4

# REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENT

**FR – 1**

- Functional Requirement (Epic) – User Requirement
- Sub Requirement (Story/Sub-Task) – Converting sign language into speech that can be understand by normal people using an application.

**FR – 2**

- Functional Requirement (Epic) – User Registration
- Sub Requirement (Story/Sub-Task) –Manual Sign up using the application or Gmail.

**FR – 3**

- Functional Requirement (Epic) – User Confirmation
- Sub Requirement (Story/Sub-Task) – OTP authentication through phone messages, email, notices, paper and confirmation.

**FR – 4**

- Functional Requirement (Epic) – Product Implementation
- Sub Requirement (Story/Sub-Task) – Install the dataset to recognise and translate hand gestures and voice for the real-time communication by using the application.

**FR - 5**

- Functional Requirement (Epic) - Payment Option
- Sub Requirement (Story/Sub-Task) - Bank transfer, Debit cards, UPI method, if pro version required.

**FR – 6**

- Functional Requirement (Epic) – Feedback Evaluation
- Sub Requirement (Story/Sub-Task) - Through the application, phone conversation and Gmail.

## 4.2 NON-FUNCTIONAL REQUIREMENT

**NFR-1**

- Non-Functional -Usability
- Requirement Description – It is used to describe the application and easy to access the application with the guidelines.

**NFR-2**

- Non-Functional - Security
- Requirement Description – It ensures the security of the application by building a firewall and two step verification support. Accessed only by authorised person by given user ID and password or OTP verification.

**NFR-3**

- Non-Functional – Reliability
- Requirement Description – To maintain the application conditions and update

the version of the application. System update and software update are possible to increase various features and durability based on technology.

**NFR-4**

- Non-Functional – Performance
- Requirement Description – This application collects the datasets of hand gestures to provide accurate prediction. Using this

method, we can communicate easily at anytime. This application is user friendly and can be accessed by both specially abled and normal people.

**NFR-5**

- Non-Functional – Availability
- Requirement Description – Depending on the requirements of the user, all

required functions will be offered. When the user requests any features, the features are made available in places where users like to know about it.

**NFR-6**

- Non-Functional – Scalability
- Requirement Description – As based on application, real-time communication is accessed on a compatible devices. The application is based on voice conversion system, hand gesture recognition and translation.

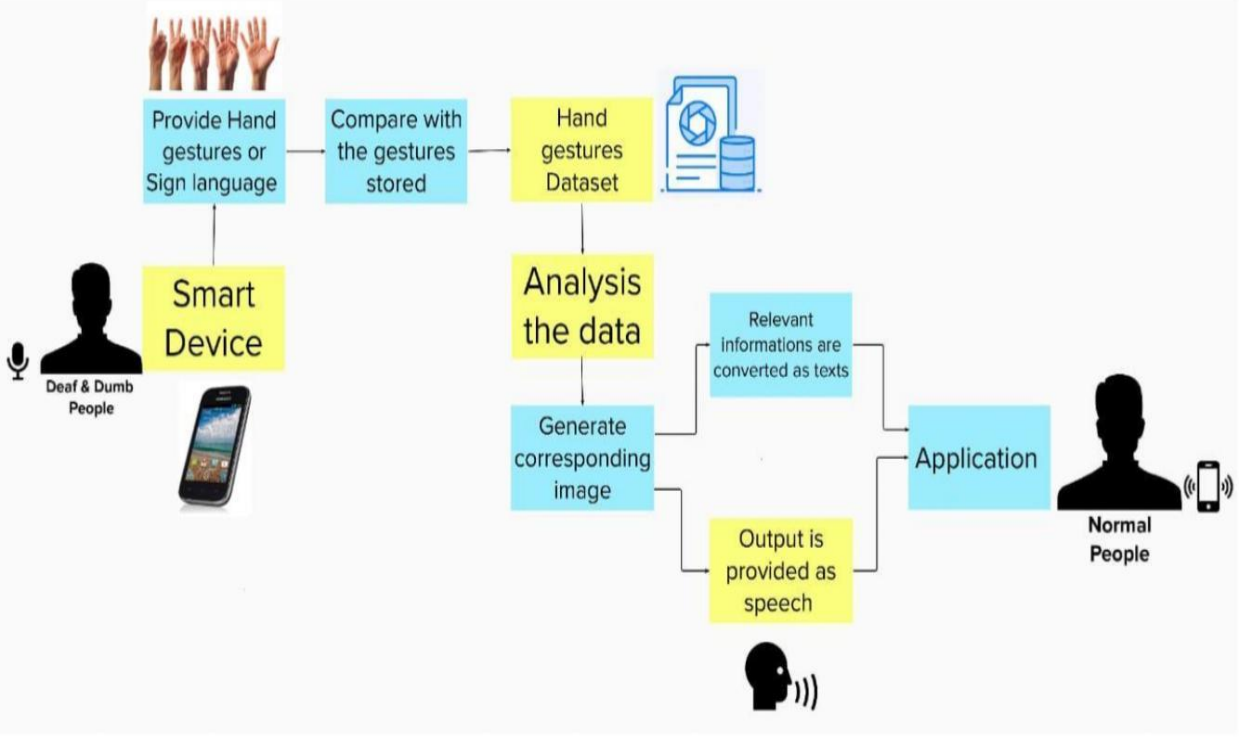
# CHAPTER 5

# PROJECT DESIGN

## 5.1 DATA FLOW DIAGRAMS

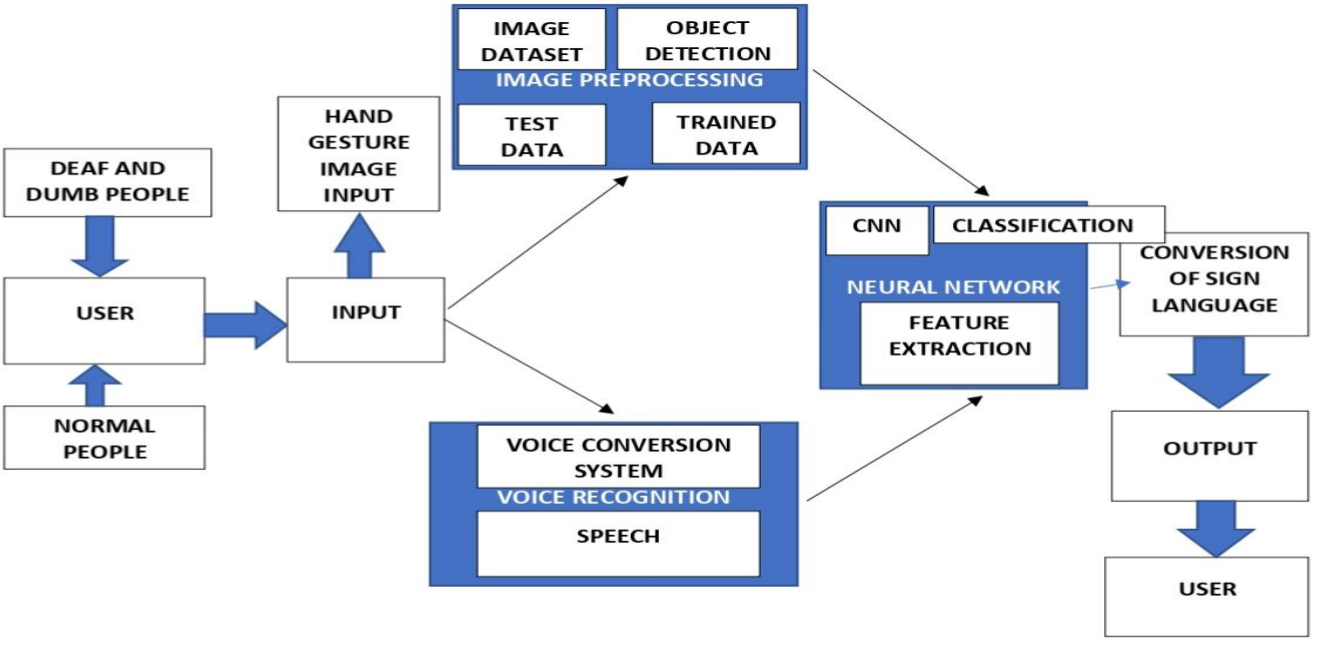
A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

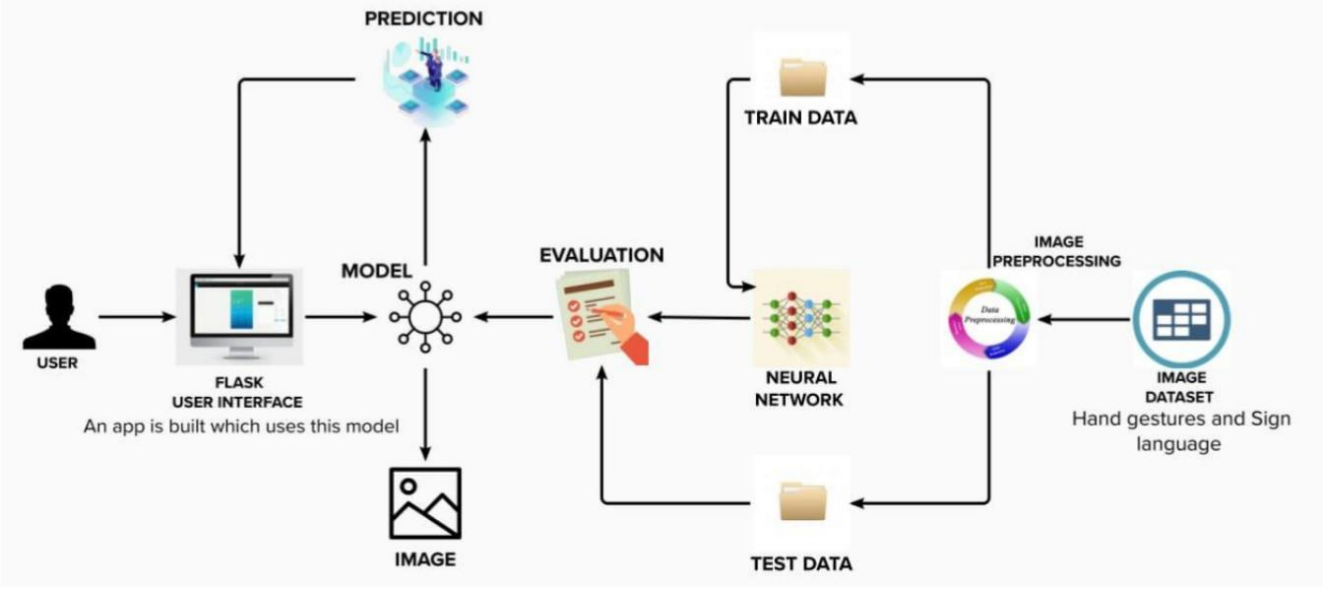


## 5.2 Solution & Technical Architecture

### Solution Architecture:




### Technical Architecture:



### 5.3 User Stories

**Sprint-01:**

**Usn-01 –** As a user, I can register for the application by entering my email, password and confirming my password

**Usn-02 -** As a user, I can see my application and made changes in any browser and register to it

**Usn-03 -** As a user, I can see my application and made changes in any browser and register to it

**Sprint-02:**

**Usn-04 -** As a user, I can register for the application by entering my email ,password, and confirmation is made.

**Sprint-01:**

**Usn-05 -** As a user, I can create my account in the application with my email and password, to get knowledge about sign language

**Usn-06 -** As a user, I can register for application by entering my email, password and confirming my password. To get details about real-time communication

**Usn-07 –** As a user, I can receive a message from the administration about conditions of application of real-time communication.

# CHAPTER 6

# PROJECT PLANNING & SCHEDULING

## 6.1 Sprint Planning & Estimation

**Sprint-01:**

**Usn-01-** Collection of Dataset

**Team members**

- SUJITH KUMAR R
- SOUNDAR SRI RAM J

**Usn-02-** Image pre-processing

**Team members**

- SRIDHAR C
- SRINATH M

**Sprint-02:**

**Usn-03-** Import required libraries, add the necessary layers and compile the model

**Team members**

- SUJITH KUMAR R
- SOUNDAR SRI RAM J

**Usn-04-** Training the image classification model using CNN

**Team members**

- SRIDHAR C
- SRINATH M

**Sprint-03:**

**Usn-05-** Training the model and testing the model's performance

**Team members**

- SUJITH KUMAR R
- SOUNDAR SRI RAM J

**Sprint-04:**

**Usn-06-** Converting the input sign language images into English alphabets.

**Team members**

- SRIDHAR C
- SRINATH M

**6.2 Sprint Delivery Schedule**

**Sprint-1**

- **Total Story Points -** 8.
- **Duration – 6** Days.
- **Sprint Start Date –** 24 Oct 2022.
- **Sprint End Date (Planned) -** 29 Oct 2022.
- **Story points Completed (as on Planned End Date)** - 5.
- **Story Release Date (Actual) -** 29 Oct 2022.

**Sprint-2**

- **Total Story Points –** 5.
- **Duration –** 6 Days.
- **Sprint Start Date –** 31 Oct 2022.
- **Sprint End Date (Planned) -** 05 Nov 2022.
- **Story points Completed (as on Planned End Date)** - 5.
- **Story Release Date (Actual) -** 05 Nov 2022.

**Sprint-3**

- **Total Story Points –** 7.
- **Duration –** 6 Days.
- **Sprint Start Date –** 07 Nov 2022.
- **Sprint End Date (Planned) -** 12 Nov 2022.
- **Story points Completed (as on Planned End Date)** - 5.
- **Story Release Date (Actual) -** 12 Nov 2022.

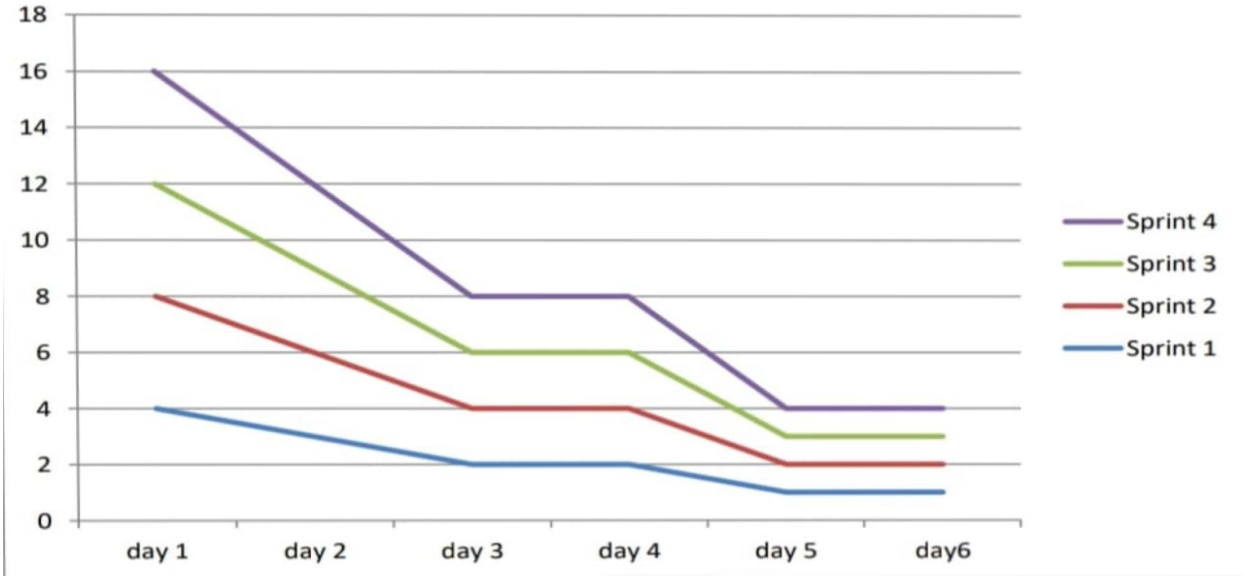
**Sprint-4**

- **Total Story Points –** 5.
- **Duration –** 6 Days.
- **Sprint Start Date –** 14 Nov 2022.
- **Sprint End Date (Planned) -** 19 Nov 2022.
- **Story points Completed (as on Planned End Date)** - 5.
- **Story Release Date (Actual) -** 19 Nov 2022.

## 6.3 Reports from JIRA

### Burndown chart report:

A burndown chart is a graphical representation of work left to do versus time and completed work. It is often used in agile software development methodologies such as scrum, jira. However burndown charts can applied to any project containing measurable time.




**Roadmap report :**

It provides the details about the project completion status ,the work yet to be completed in four ways like days, months, weeks, quarters.

# CHAPTER 7

# CODING & SOLUTIONING

## 7.1 Feature 1

The user can choose which sign language to read on the different sign language standards that exist.

**MODEL BUILDING**

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
# Creating Model
model=Sequential()
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
# Adding Hidden Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
# Adding Output Layer
model.add(Dense(9,activation='softmax'))
# Compiling the Model
 model.compile(loss='categorical_crossentropy',optimizer= 'adam',metrics= ['accuracy'])
 # Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data =x_test,validation_steps=len(x_test))
```

**C:\Users\sujit\AppData\Local\Temp\ipykernel_6864\1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated will be removed in a future version. Please use `Model.fit`, which supports generators. model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))**

**Epoch 1/10 18/18 [==============================] - 948s 49s/step - loss: 1.2305 - accuracy: 0.6189 - val_loss: 0.4374 - val_accuracy: 0.9062 Epoch 2/10 18/18 [==============================] - 369s 20s/step - loss: 0.2928 - accuracy: 0.9173 - val_loss: 0.2673 - val_accuracy: 0.9373 Epoch 3/10 18/18 [==============================] - 219s 12s/step - loss: 0.1229 - accuracy: 0.9675 - val_loss: 0.1901 - val_accuracy: 0.9658 Epoch 4/10 18/18 [==============================] - 180s 10s/step - loss: 0.0699 - accuracy: 0.9823 - val_loss: 0.1987 - val_accuracy: 0.9698 Epoch 5/10 18/18 [==============================] - 164s 9s/step - loss: 0.0445 - accuracy: 0.9880 - val_loss: 0.1843 - val_accuracy: 0.9742 Epoch 6/10 18/18 [==============================] - 159s 9s/step - loss: 0.0299 - accuracy: 0.9917 - val_loss: 0.1883 - val_accuracy: 0.9760 Epoch 7/10 18/18 [==============================] - 162s 9s/step - loss: 0.0232 - accuracy: 0.9942 - val_loss: 0.2062 - val_accuracy: 0.9751 Epoch 8/10 18/18 [==============================] - 163s 9s/step - loss: 0.0170 - accuracy: 0.9959 - val_loss: 0.2146 - val_accuracy: 0.9764 Epoch 9/10 18/18 [==============================] - 177s 10s/step - loss: 0.0131 - accuracy: 0.9975 - val_loss: 0.2212 - val_accuracy: 0.9764 Epoch 10/10 18/18 [==============================] - 358s 20s/step - loss: 0.0083 - accuracy: 0.9987 - val_loss: 0.2394 - val_accuracy: 0.9764**

model.save('asl_model_84_54.h5')

**# Current accuracy is 0.8454**

**TEST THE MODEL:**

```
model.save('asl_model_84_54.h5')
# Current accuracy is 0.8454
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('asl_model_84_54.h5')
img=image.load_img(r'C:\Users\sujit\Desktop\PROJECT\DATA_COLLECTION\
test_set\D\2.png',target_size=(64,64))
img
```

```
x=image.img_to_array(img)
x.ndim
```

**3**

```
x=np.expand_dims(x,axis=0)
x.ndim
```

**4**

```
pred=np.argmax(model.predict(x),axis=1)
```

**1/1 [==============================] - 8s 8s/step**

```
pred
```

```
array([3], dtype=int64)

index=['A','B','C','D','E','F','G','H','I']
print(index[pred[0]])
```

**D**

### 7.2 Feature 2

The communication gap between deaf and dumb people and the general public can be bridged with a mobile application.

**MOBILE APP:**

```
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()
```

# CHAPTER 8
# TESTING

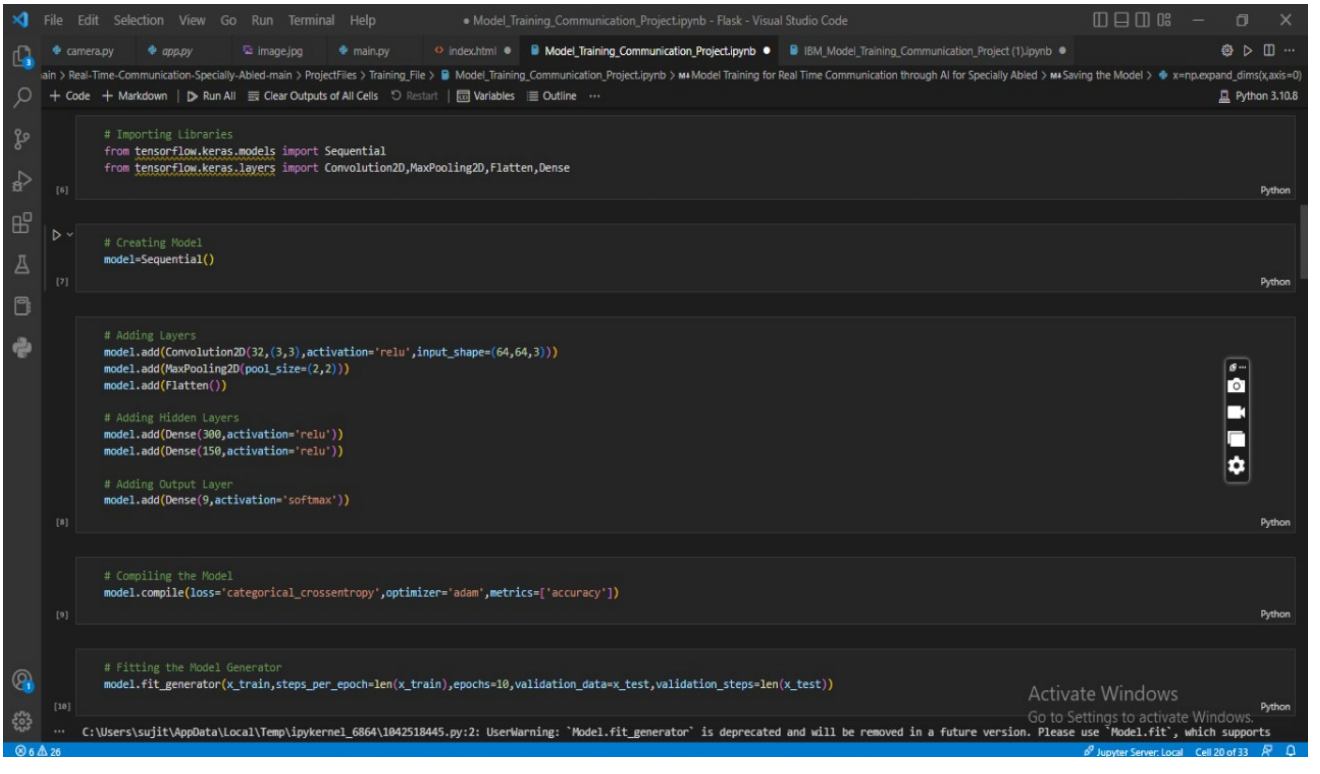
## 8.1 Test Cases

- Our code was tested on various angle to check whether it gives the correct output.
- To satisfy the customer's expectations, we tested it fully.

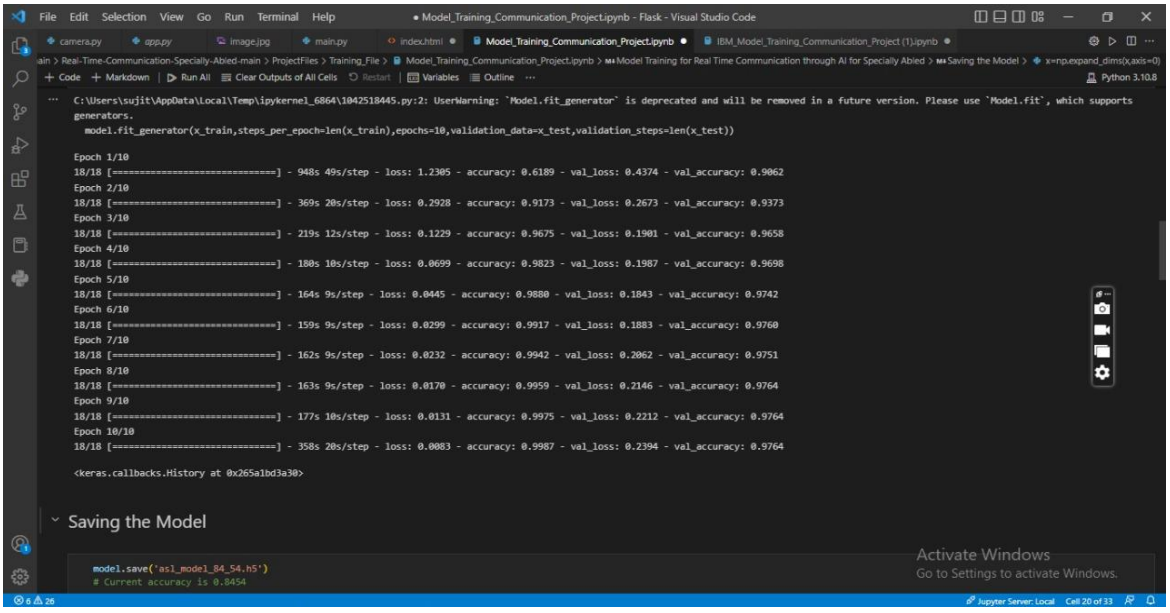
## 8.2 User Acceptance Testing

Our project was tested by an end user to verify that it has working correctly.

- **Model Summary**



- **Accuracy**




Training Accuracy –99.6%

Validation Accuracy –98.3%

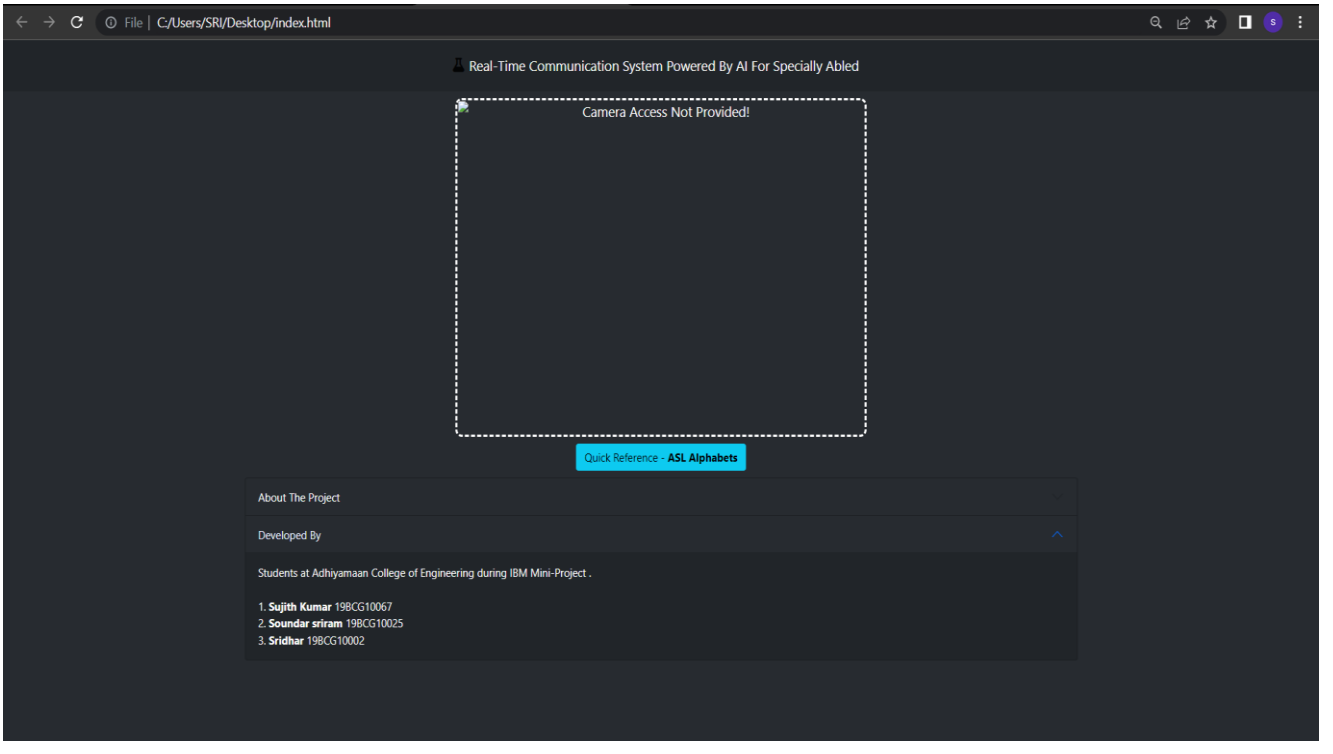
# CHAPTER 9

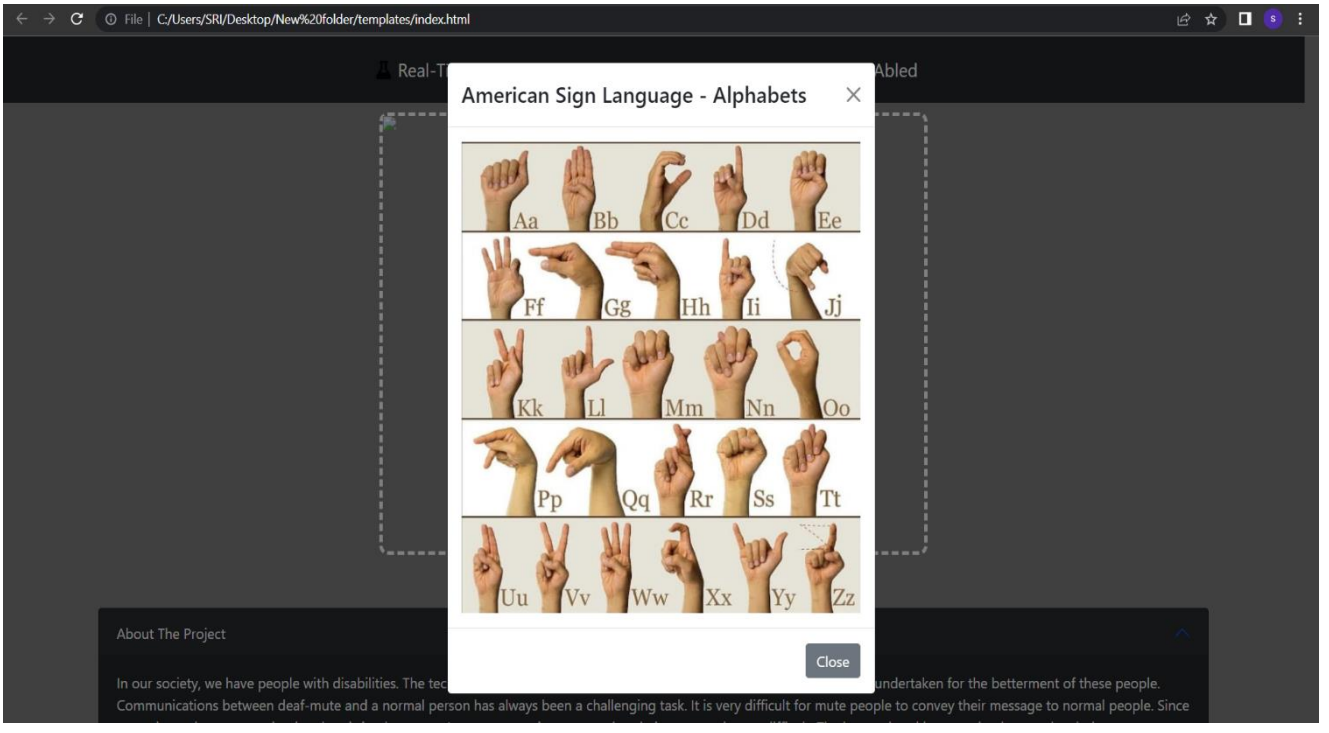
# RESULTS

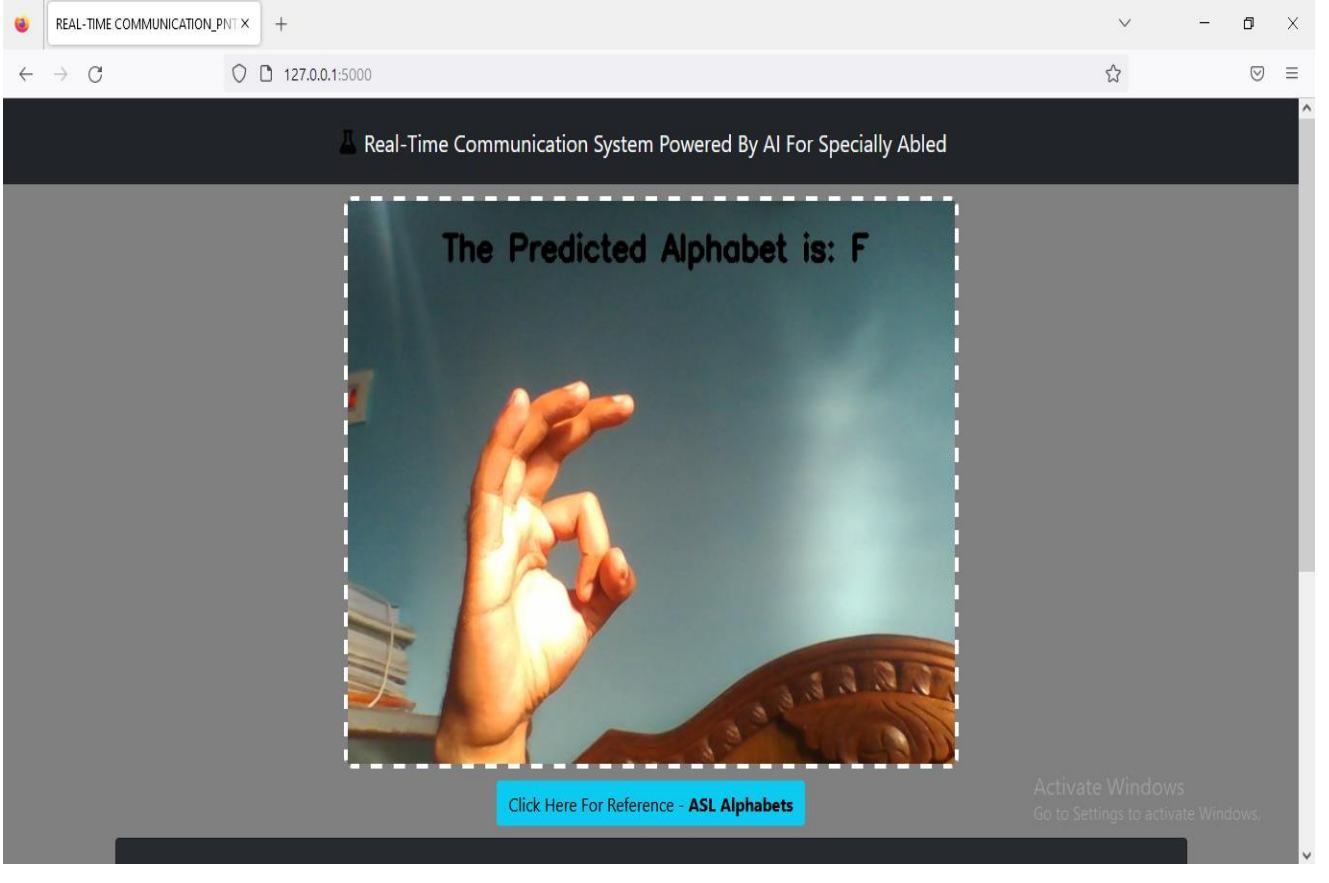
## 9.1 Performance Metrics

- 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 recognise the equivalent Alphabet is shown on the screen.

Some sample images of the output are provided below:




File | C:/Users/SRI/Desktop/New%20folder/templates/index.html
American Sign Language - Alphabets
Aa Bb Cc Dd Ee
Ff Gg Hh Ii Jj
Kk Ll Mm Nn Oo
Pp Qq Rr Ss Tt
Uu Vv Ww Xx Yy Zz
Close
About The Project


REAL-TIME COMMUNICATION_PNT
127.0.0.1:5000
Real-Time Communication System Powered By AI For Specially Abled
The Predicted Alphabet is: F
Click Here For Reference - ASL Alphabets
Activate Windows
Go to Settings to activate Windows.

# CHAPTER 10

## ADVANTAGES & DISADVANTAGES

**Advantages:**

1.It is possible to create a mobile application to bridge the communication gap betwee n 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.

# CHAPTER 11

# CONCLUSION

## 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.

# CHAPTER 12

# FUTURE SCOPE

## Future Scope:

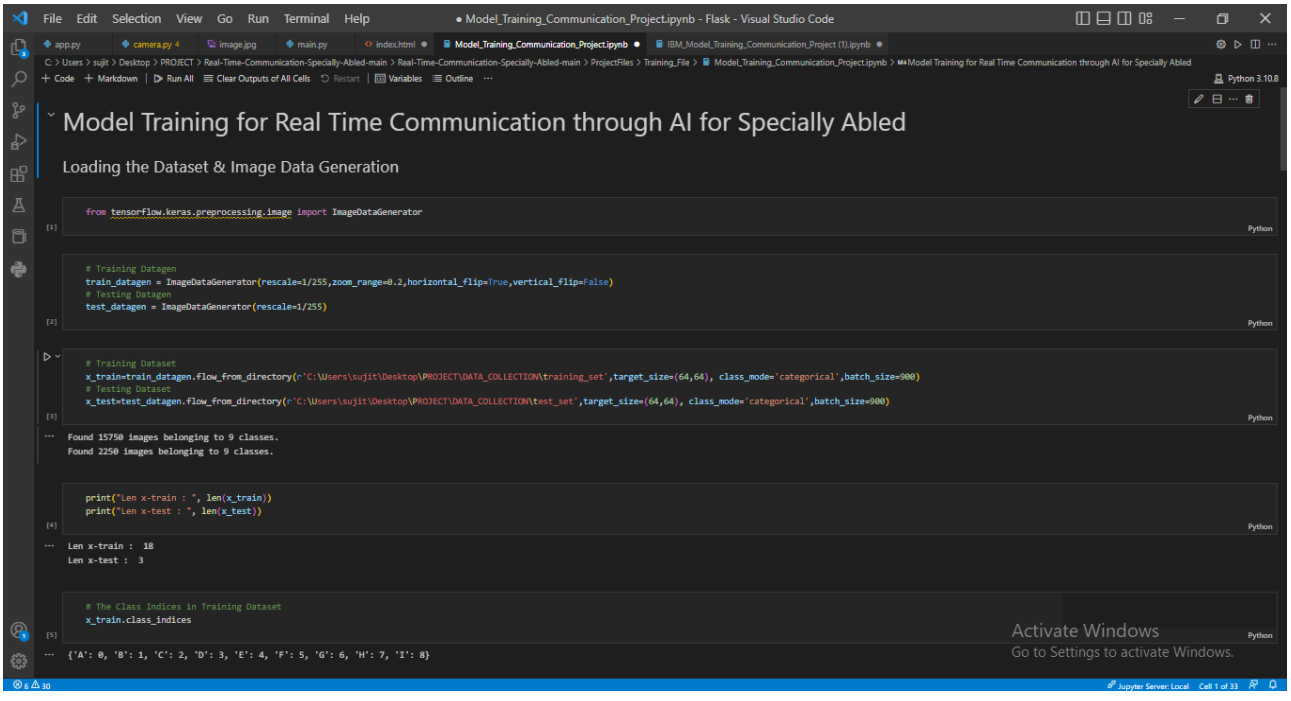
Having a technology that can translate hand sign language to its corresponding alphab et is a game changer in the field of communication and AI
for the specially abled people
such as deaf and dumb. With introduction of gesture recognition, the web app can easily be Expanded to recognize letters beyond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

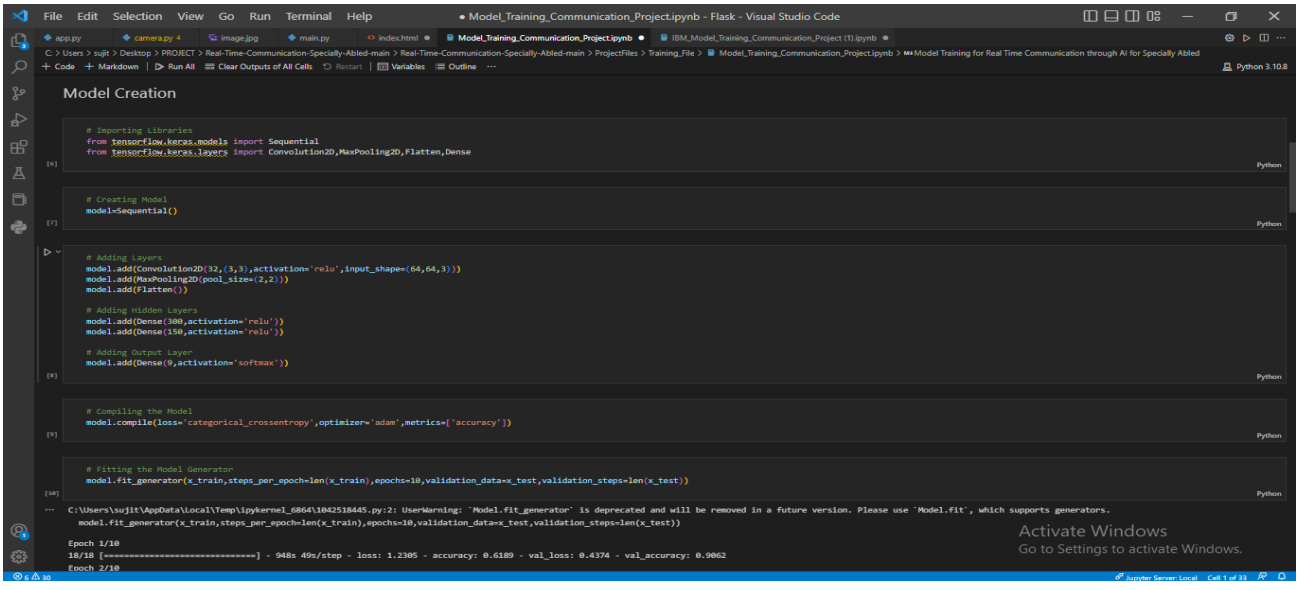
# CHAPTER 13

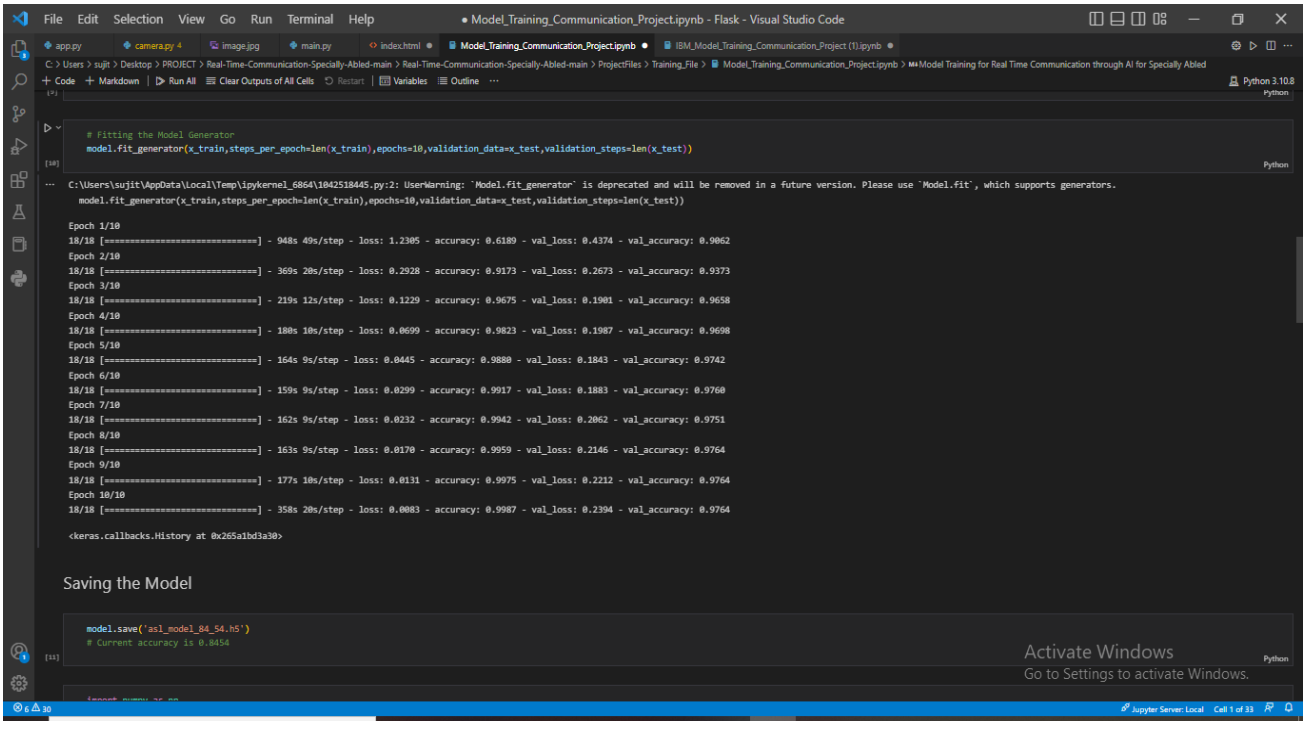
# APPENDIX

## Appendix:

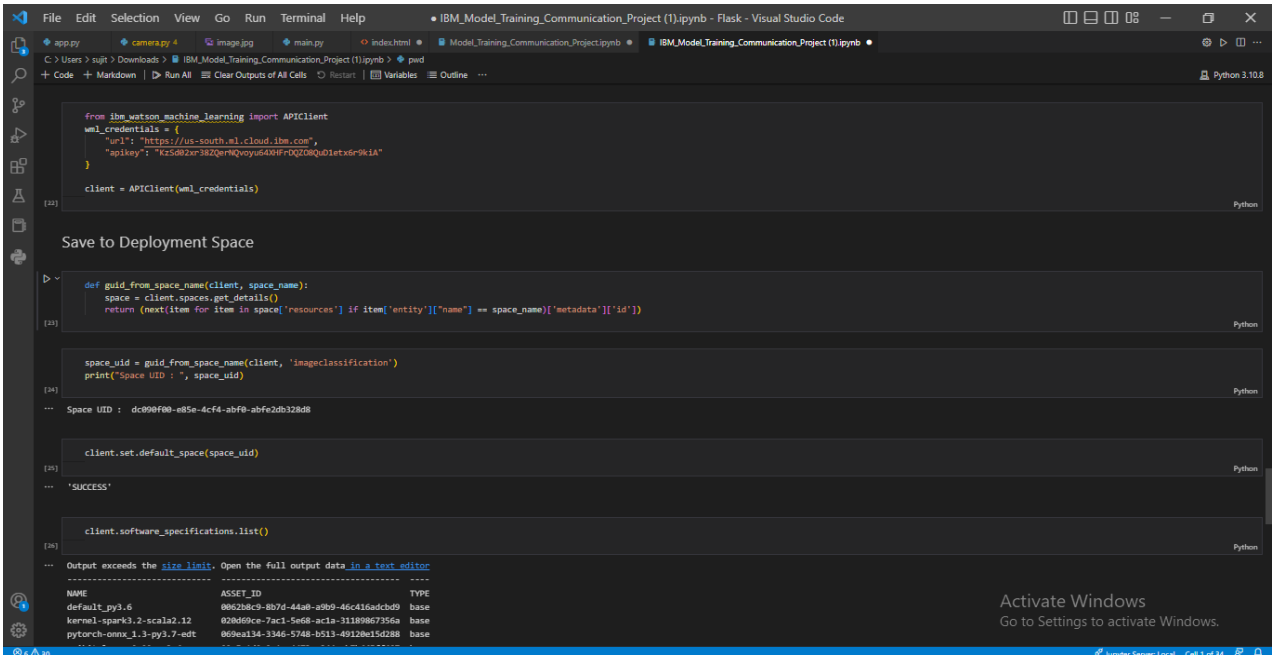
## Source Code for model training and saving:



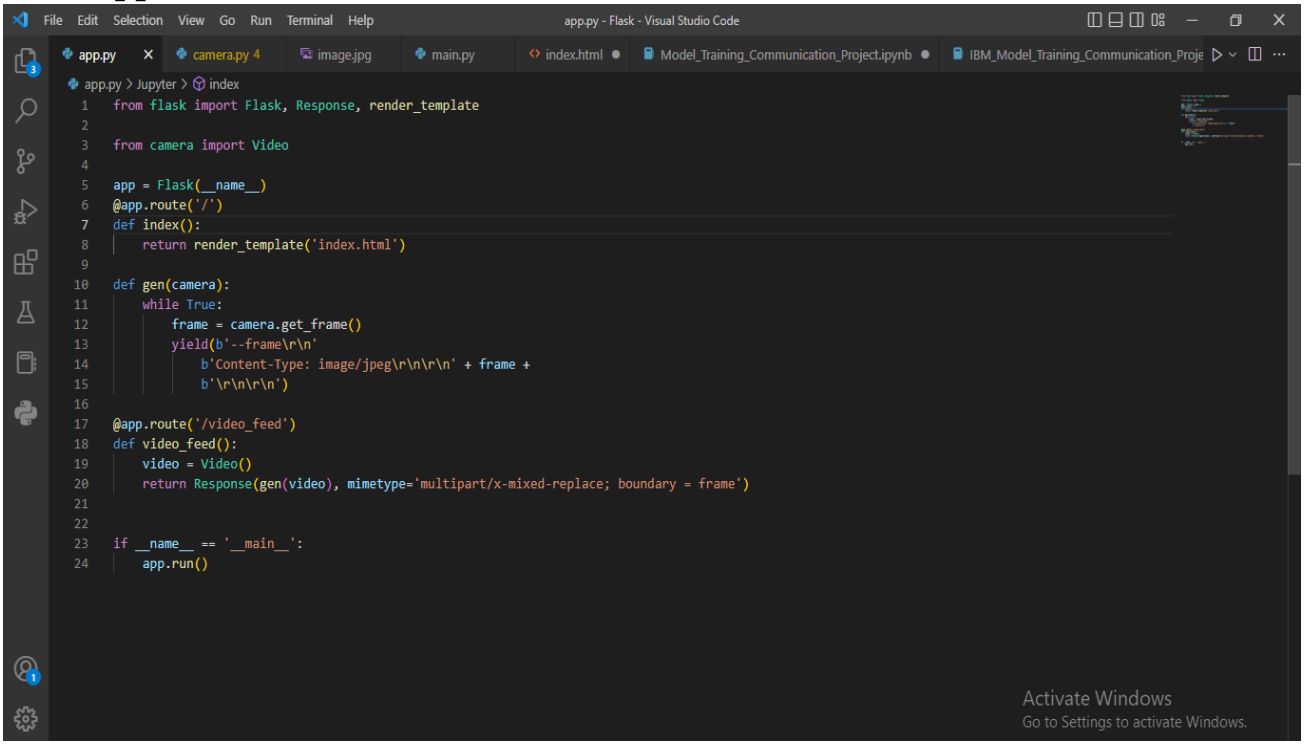

File Edit Selection View Go Run Terminal Help
• Model_Training_Communication_Project.ipynb - Flask - Visual Studio Code
Model Creation
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
# Creating Model
model=Sequential()
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
# Adding Hidden Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
# Adding Output Layer
model.add(Dense(9,activation='softmax'))
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
C:\Users\sujit\AppData\Local\Temp\ipykernel_6864\1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
Epoch 1/10
18/18 [==============================] - 948s 49s/step - loss: 1.2305 - accuracy: 0.6189 - val_loss: 0.4374 - val_accuracy: 0.9062
Epoch 2/10
Activate Windows
Go to Settings to activate Windows.

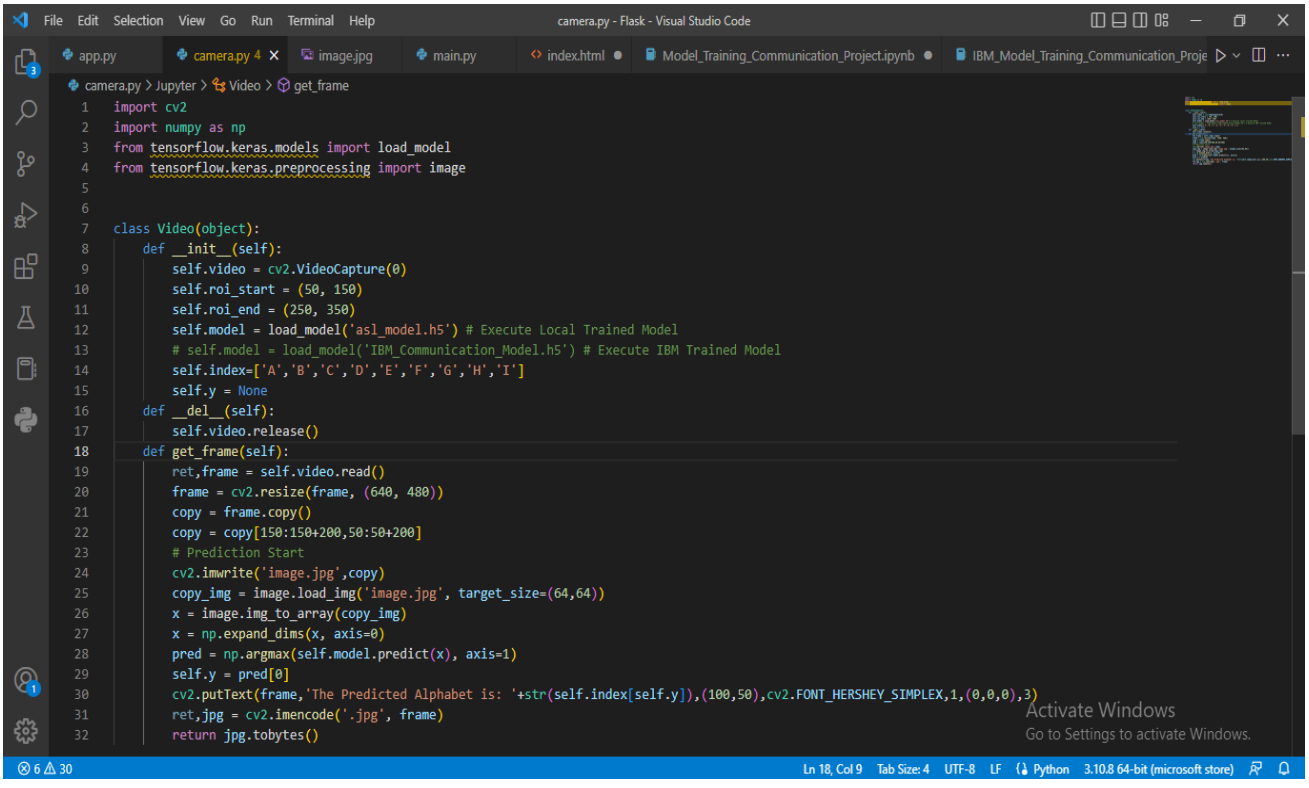

File Edit Selection View Go Run Terminal Help
• Model_Training_Communication_Project.ipynb - Flask - Visual Studio Code
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
C:\Users\sujit\AppData\Local\Temp\ipykernel_6864\1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
Epoch 1/10
18/18 [==============================] - 948s 49s/step - loss: 1.2305 - accuracy: 0.6189 - val_loss: 0.4374 - val_accuracy: 0.9062
Epoch 2/10
18/18 [==============================] - 369s 20s/step - loss: 0.2928 - accuracy: 0.9173 - val_loss: 0.2673 - val_accuracy: 0.9373
Epoch 3/10
18/18 [==============================] - 219s 12s/step - loss: 0.1229 - accuracy: 0.9675 - val_loss: 0.1901 - val_accuracy: 0.9658
Epoch 4/10
18/18 [==============================] - 180s 10s/step - loss: 0.0699 - accuracy: 0.9823 - val_loss: 0.1987 - val_accuracy: 0.9698
Epoch 5/10
18/18 [==============================] - 164s 9s/step - loss: 0.0445 - accuracy: 0.9880 - val_loss: 0.1843 - val_accuracy: 0.9742
Epoch 6/10
18/18 [==============================] - 159s 9s/step - loss: 0.0299 - accuracy: 0.9917 - val_loss: 0.1883 - val_accuracy: 0.9760
Epoch 7/10
18/18 [==============================] - 162s 9s/step - loss: 0.0232 - accuracy: 0.9942 - val_loss: 0.2062 - val_accuracy: 0.9751
Epoch 8/10
18/18 [==============================] - 163s 9s/step - loss: 0.0170 - accuracy: 0.9959 - val_loss: 0.2146 - val_accuracy: 0.9764
Epoch 9/10
18/18 [==============================] - 177s 10s/step - loss: 0.0131 - accuracy: 0.9975 - val_loss: 0.2212 - val_accuracy: 0.9764
Epoch 10/10
18/18 [==============================] - 358s 20s/step - loss: 0.0083 - accuracy: 0.9987 - val_loss: 0.2394 - val_accuracy: 0.9764
<keras.callbacks.History at 0x265a1bd3a30>
Saving the Model
model.save('asl_model_84_54.h5')
# Current accuracy is 0.8454
Activate Windows
Go to Settings to activate Windows.

# IBM Model Training and Download Code:




# Web App Code:

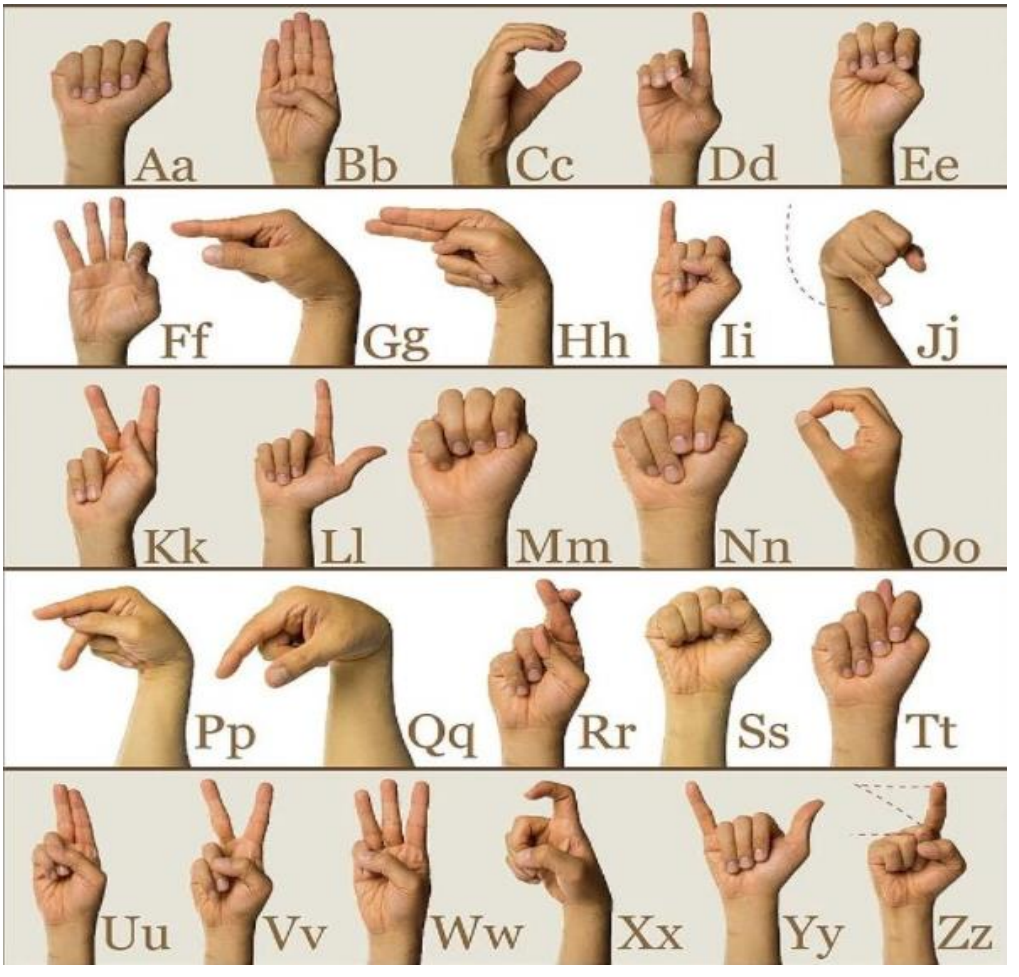





## The Generated API Key:

```
"url": "https://us-south.ml.cloud.ibm.com",
"apikey": "KzSd02xr38ZQerNQvoyu64XHFrDQZO8QuD1etx6r9kiA"
```

## American Sign Language Standard Reference:




**GitHub & Project Demo Link**

**Project Demo Link:**

**https://drive.google.com/file/d/1zZvkRsKLtv47ezPEvlVdN-1BTOvh9QLr/view?usp=drivesdk**

**GitHub Project Link:**

**https://github.com/IBM-EPBL/IBM-Project-12414-1659450858**