**PROJECT REPORT**

|  |  |  |
| --- | --- | --- |
| **Project Title** | : | Real-Time Communication System Powered  By AI for Specially Abled |
|  |  |  |

**Project Members** :-

Siddarth V S

Mohanraam V K

Rupesh Kanna S

Sharan Kumar J A S

Harish Kumar S

**1) INTRODUCTION**

* 1. **PROJECT OVERVIEW**

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

* 1. **PURPOSE OF THE PROJECT**

The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

**2)** **LITERATURE SURVEY**

* 1. **EXISTING PROBLEM**

There are handicapped people in our society. Although technology is constantly evolving, little is being done to improve the lives of these people. It has always been difficult to communicate with someone who is deaf-mute. It is quite challenging for silent persons to communicate with non-mute people because hand sign language is not taught to the general public. It might be quite challenging for them to communicate at times of crisis. In circumstances where other modes of

communication, like speech, are not possible, the human hand has remained a common alternative for information transmission. To have a proper communication between a normal person and a handicapped person in any language, a voice conversion system with hand gesture recognition and translation will be very helpful.

* 1. **REFERENCES**

[1] Thoutam, N., Jha, D. K., Jaiswal, L., Deshmukh, S., & Raj, R. Hand gesture, Text and Speech Translation and Recognition System for specially abled people using AI.

[2] Shinde, Shweta S., Rajesh M. Autee, and Vitthal K. Bhosale. "Real time two way communication approach for hearing impaired and dumb person based on image processing." Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference on. IEEE, 2016.

[3] Shangeetha, R. K., V. Valliammai, and S. Padmavathi. "Computer vision based approach for Indian Sign Language character recognition." Machine Vision and Image Processing (MVIP), 2012 International Conference on. IEEE, 2012.

[4] Sood, Anchal, and Anju Mishra. "AAWAAZ: A communication system for deaf and dumb." Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 2016 5th International Conference on. IEEE, 2016.

[5] Ahire, Prashant G., et al. "Two Way Communicator between Deaf and Dumb People and Normal People." Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on. IEEE, 2015.

[6] Ms R. Vinitha and Ms A. Theerthana. "Design And Development Of Hand Gesture Recognition System For Speech Impaired People."

[7] Kusurnika Krori Dutta, Satheesh Kumar Raju K, Anil Kumar G S,Sunny Arokia Swarny B,“Double handed Indian Sign Language to speech and text”, IEEE, 2015 Third International Conference on ImageInformation Processing.

[8] Aarthi M, Vijayalakshmi P, “Sign language to speech conversion”,IEEE,2016 fifth international conference on recent trends in information technology.

[9] ‘‘Malay Sign Language Gesture Recognition System ”Tan TianSwee, Sh-Hussain Salleh, A.K.Ariff, Chee-Ming Ting, Siew KeanSeng, and Leong Seng Huat, ‘‘International Conference on Intelligent And Advanced Systems 2007”

[10] Sarth Pham The Hai, Huynh Chau Thinh, Bui Van Phuc, Ha HoangKha, "Automatic feature extraction for Vietnamese sign language recognition using support vector machine", Recent Advances in Signal Processing Communications & Computing (SigTelCom)2018 2nd International Conference on, pp. 146-151, 2018.

* 1. **PROBLEM STATEMENT DEFINITION**

Communication is the medium by which we can share our thoughts or convey the

messages with other person. But for specially abled people like dumb and deaf people

this is challenging. Gesture shows an expressive movement of body which convey some

message but not everyone is aware of sign language. Since deaf and dumb people have to

depend on some sort of visual communication. In recent years, there has been so many

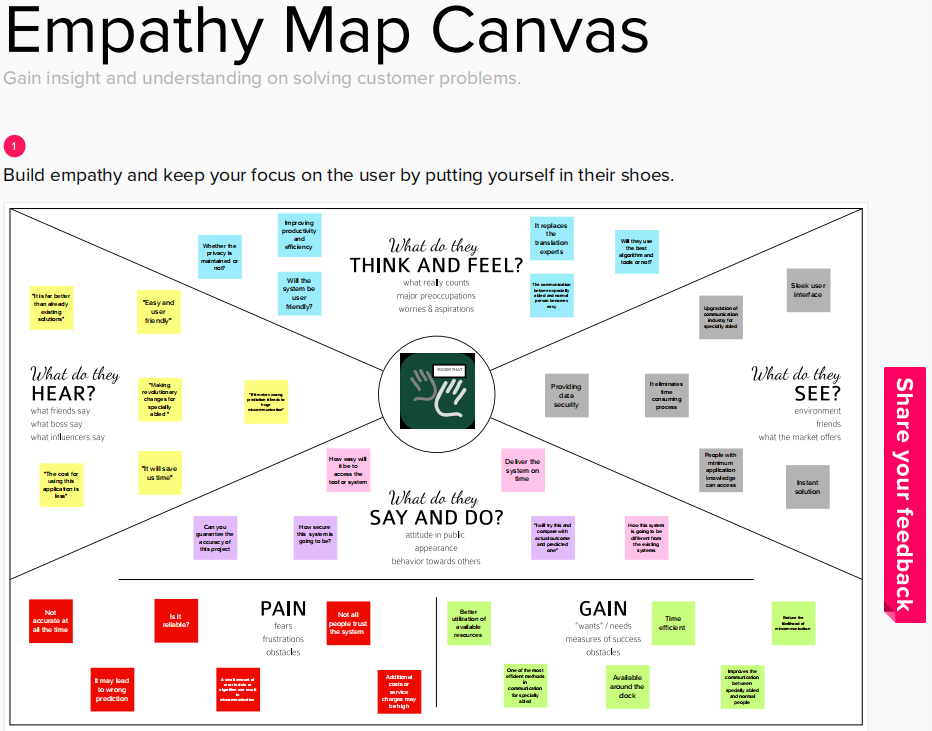
advancement in technology. Gesture recognition is the mathematical interpretation of a

human motion by a computing device. Sign language provide best communication

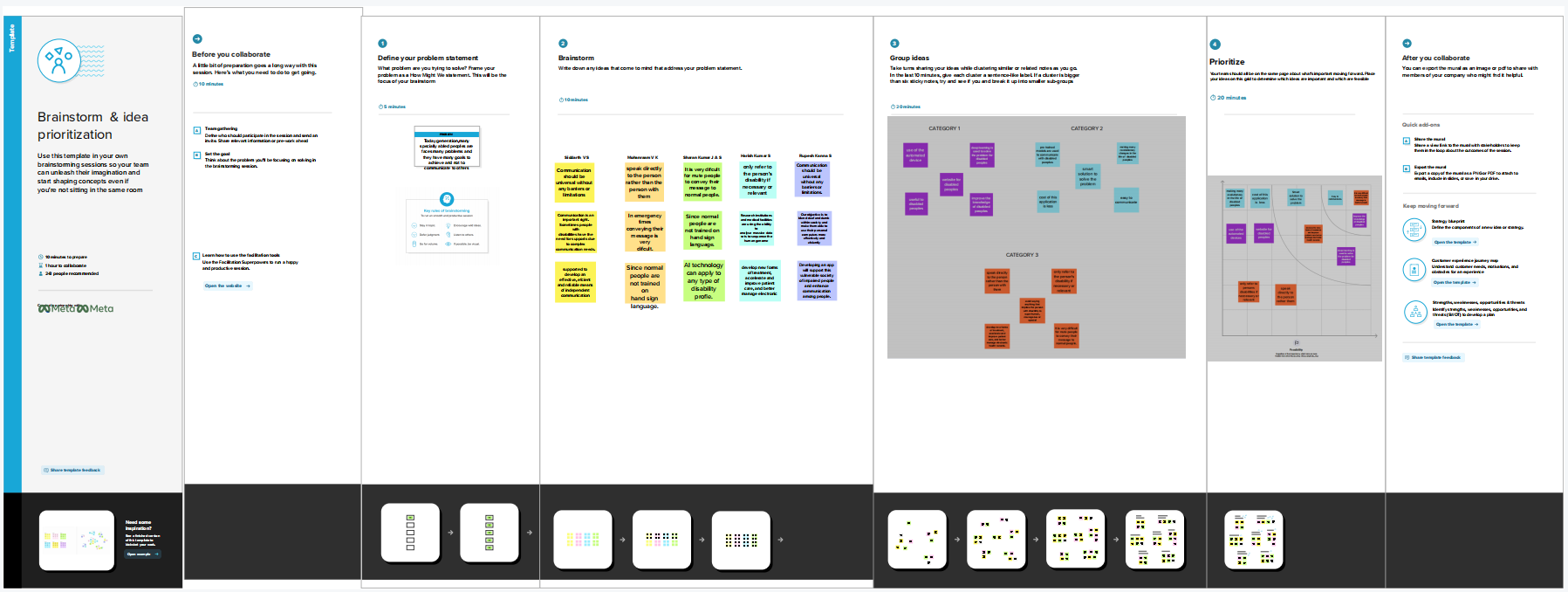
platform for the hearing impaired and dumb person to communicate with normal person.

**3)** **IDEATION AND PROPOSED SOLTION**

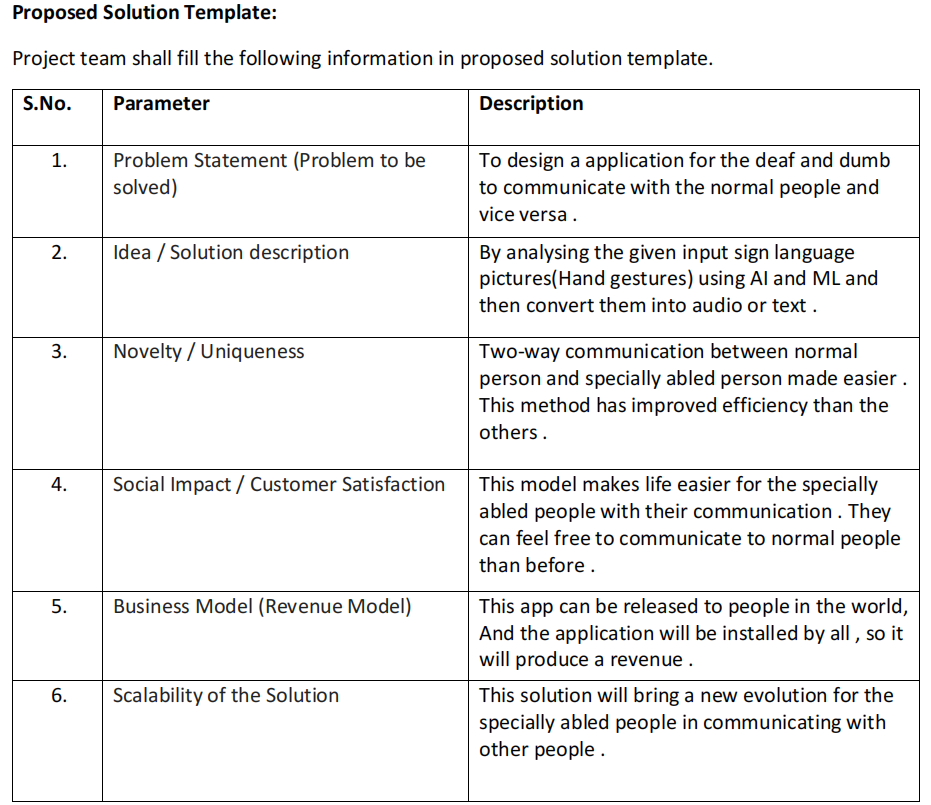
**3.1)** **EMPATHY MAP CANVAS**



**3.2) IDEATION AND BRAINSTORMING**



**3.3) PROPOSED SOLUTION**

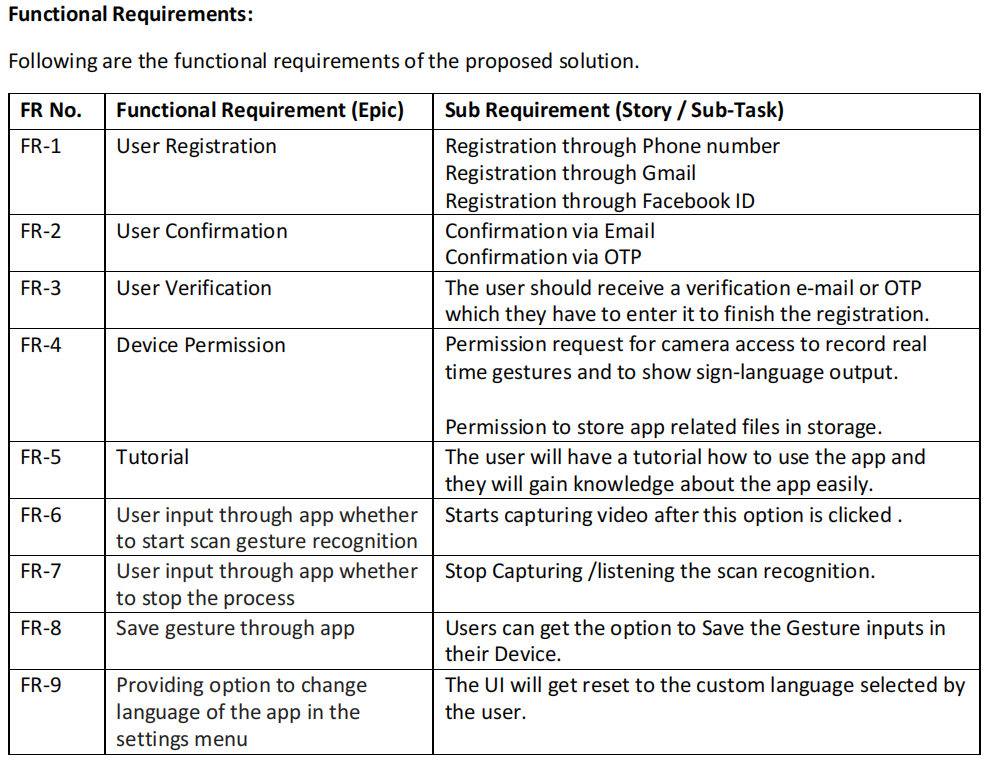


**3.4) PROBLEM SOLUTION FIT**

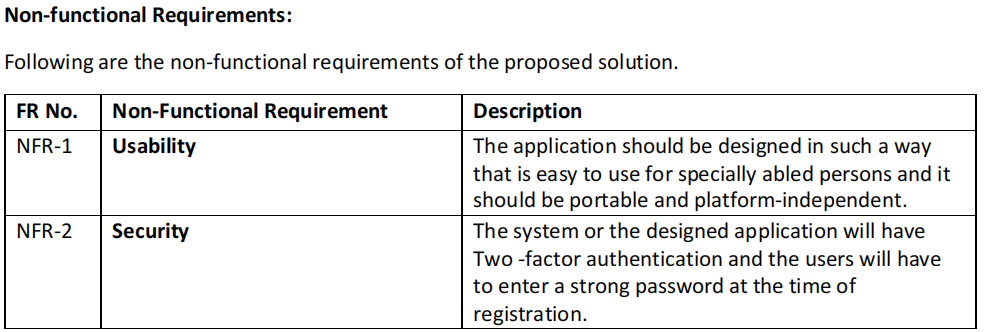


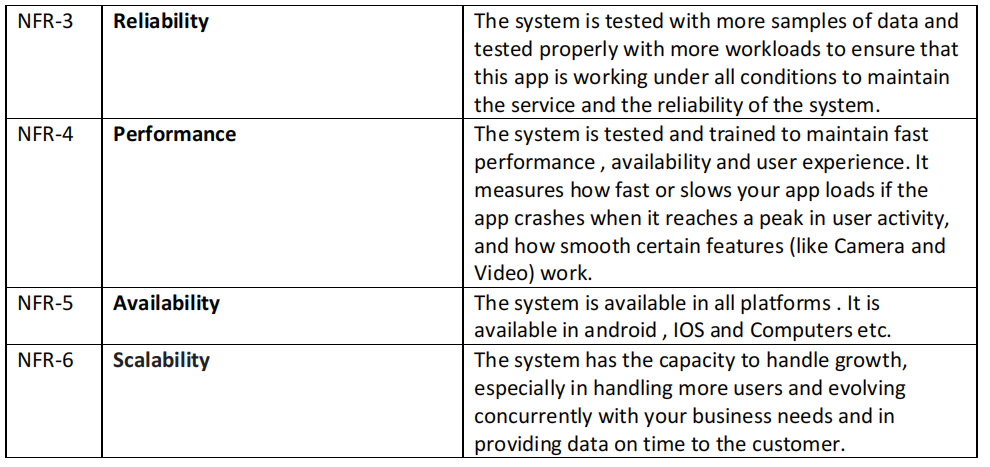
**4) REQUIREMENT ANALYSIS**

**4.1) FUNCTIONAL REQUIREMENTS**



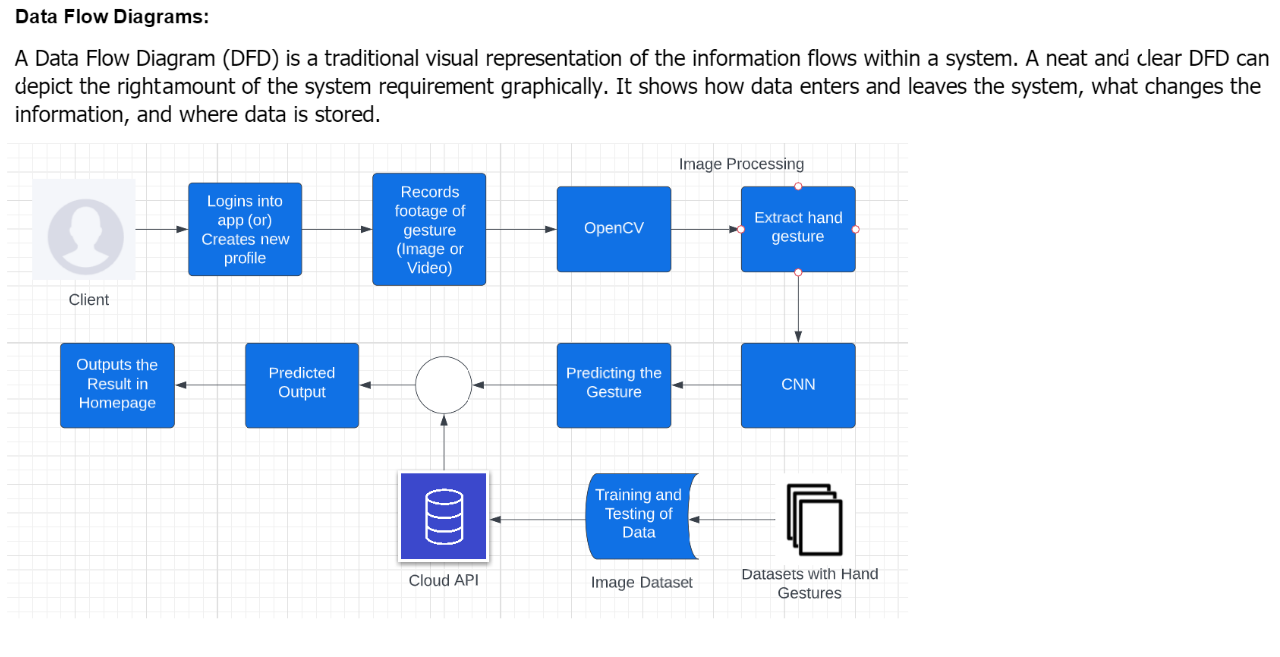
**4.2) NON FUNCTIONAL REQUIREMENTS**



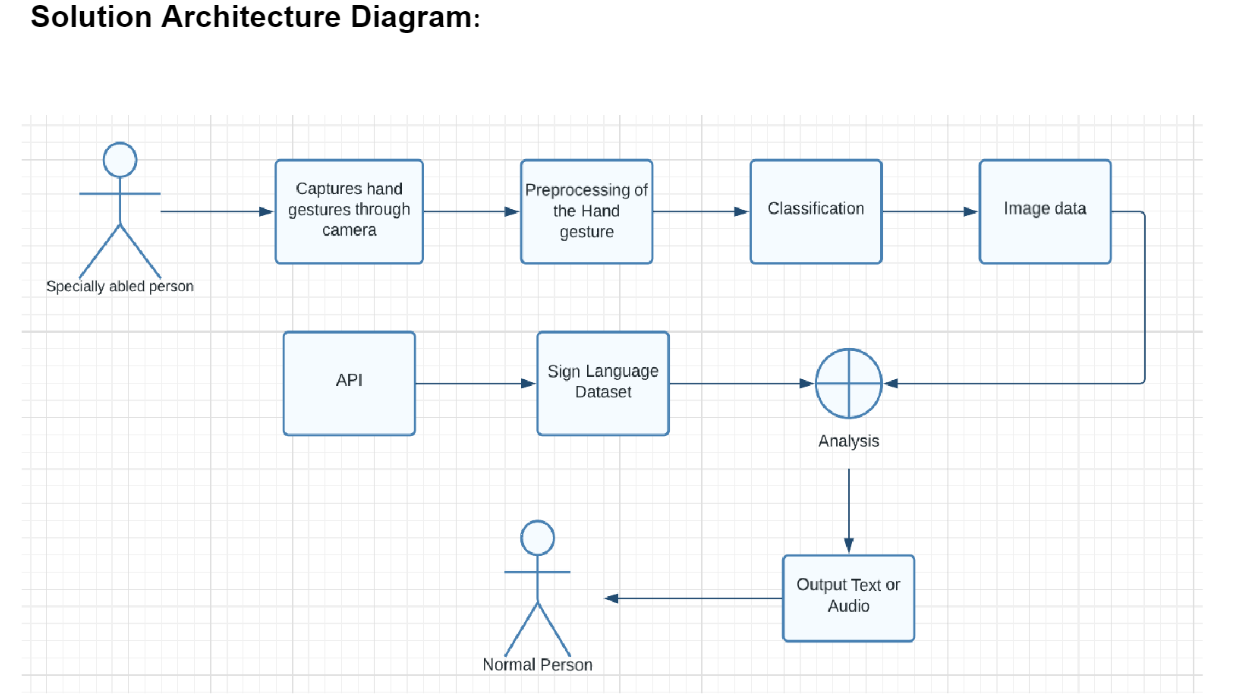


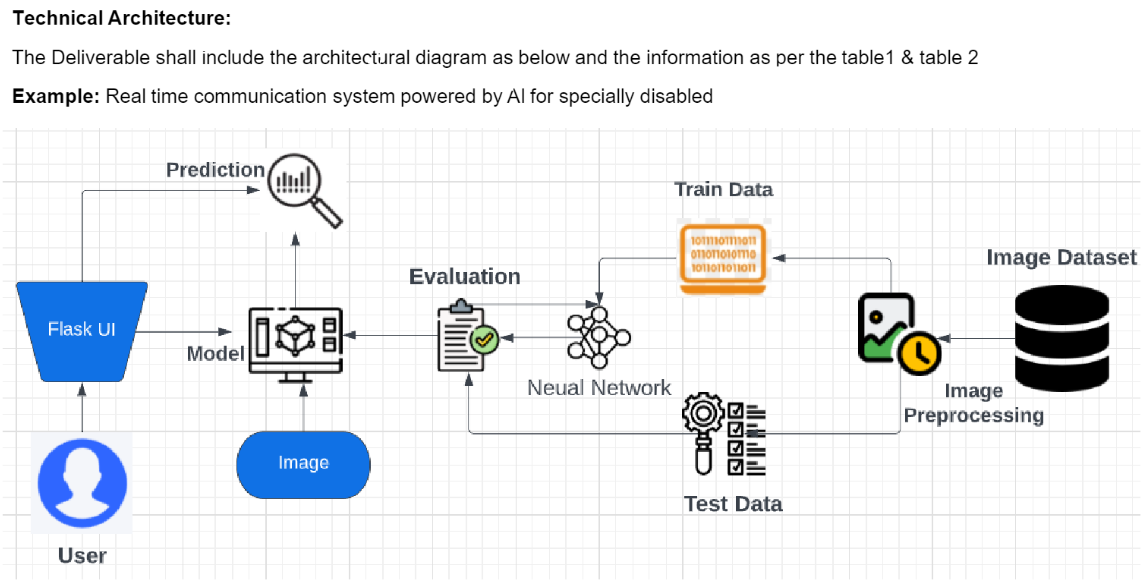
**5) PROJECT DESIGN**

**5.1) DATA FLOW DIAGRAMS**

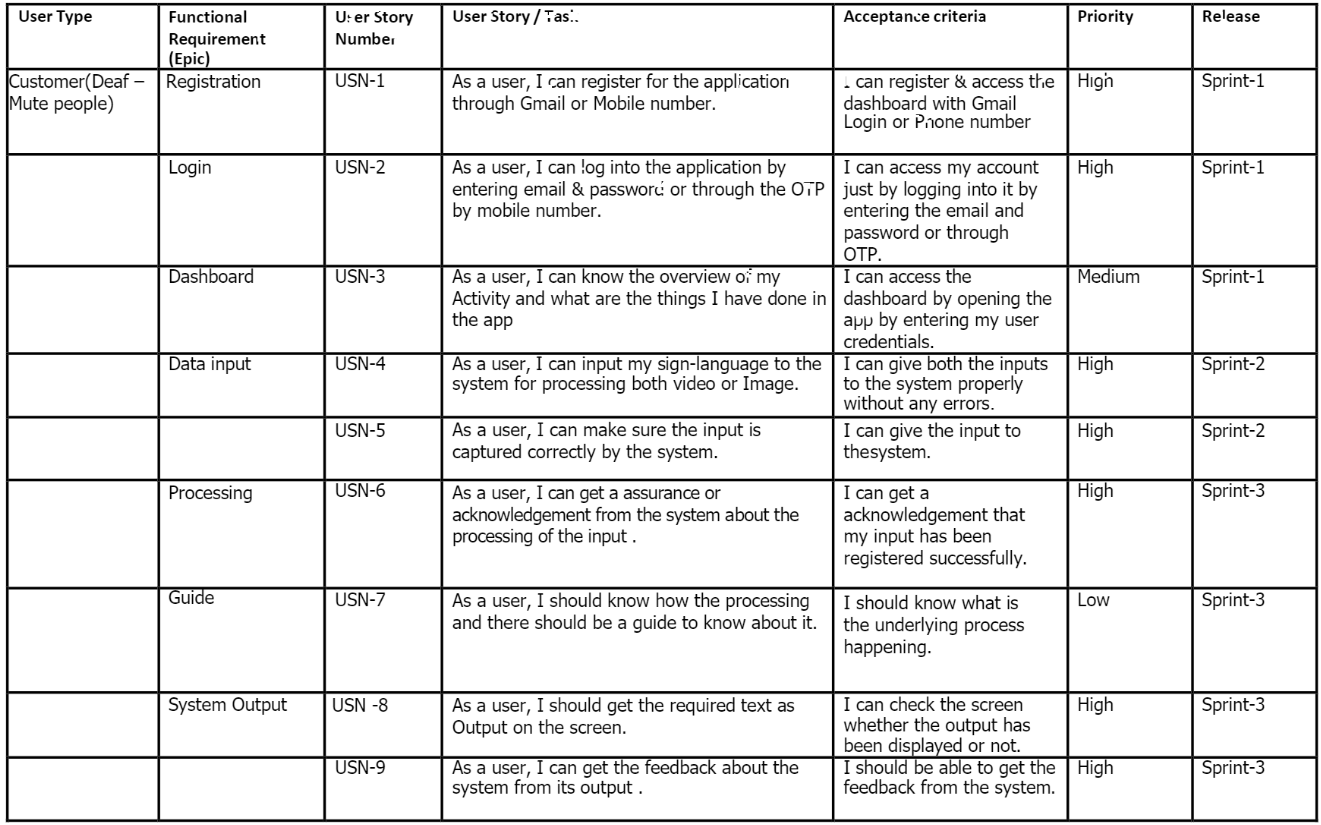


**5.2) SOLUTION AND TECHNICAL ARCHITECTURE**



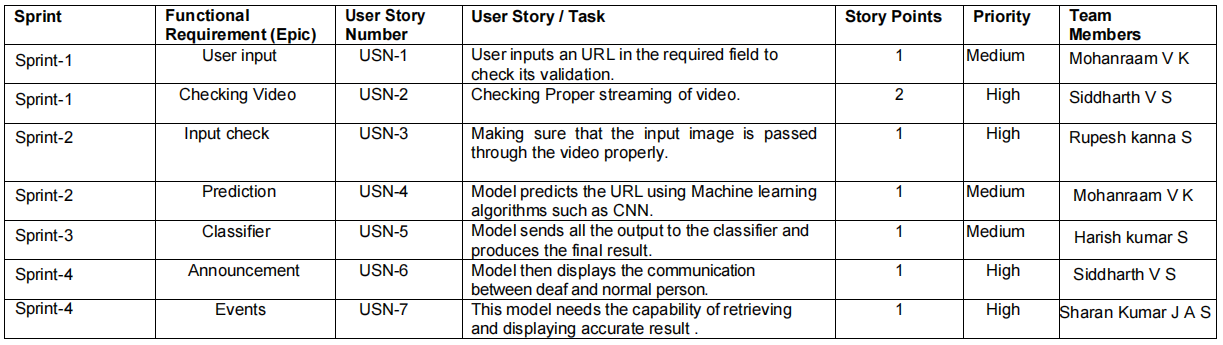


**5.3) USER STORIES**



**6) PROJECT PLANNING AND SCHEDULING**

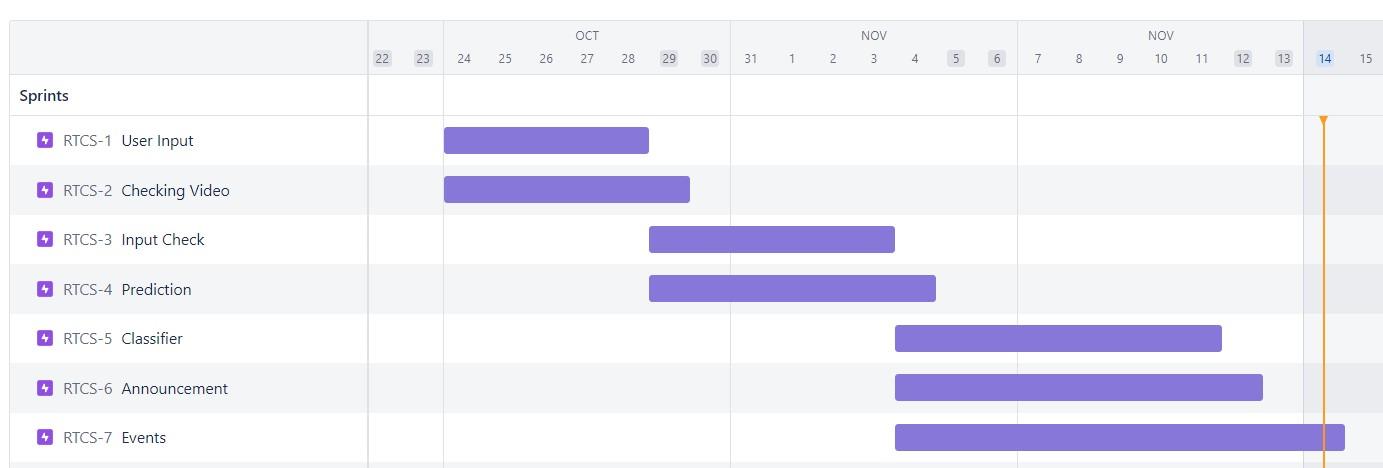
**6.1) SPRINT PLANNING AND ESTIMATION**



**6.2) SPRINT DELIVERY SCHEDULE**



**6.3) REPORTS FROM JIRA**



**7) CODING AND SOLUTIONING**

**7.1) Feature 1**

We added numbers and alphabets hand signs as image inputs using hand signs

The below code is used for this feature

**Main code**

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense

train\_data=ImageDataGenerator(rescale=1./255,zoom\_range=0.2,horizontal\_flip=True)

test\_data=ImageDataGenerator(rescale=1./255,validation\_split=0.5)

xtrain=train\_data.flow\_from\_directory('E:/Projects/Jyupter/Dataset/Hand Sign/Train',

target\_size=(64,64),class\_mode='categorical',batch\_size=100)

xtest=test\_data.flow\_from\_directory('E:/Projects/Jyupter/Dataset/Hand Sign/Test',

target\_size=(64,64),class\_mode='categorical',batch\_size=100)

#Layering

model=Sequential()

model.add(Convolution2D(32,(3,3),activation='relu',input\_shape=(64,64,3)))

model.add(MaxPooling2D(pool\_size=(2,2)))

model.add(Flatten())

model.add(Dense(300,activation='relu'))

model.add(Dense(150,activation='relu'))

model.add(Dense(36,activation='softmax'))

#Compile

model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy'])

#Fit

model.fit(xtrain,steps\_per\_epoch=len(xtrain),

epochs=25,validation\_data=xtest,validation\_steps=len(xtest))

model.save('Hand-SignV2.h5')

#Test

from tensorflow.keras.preprocessing import image

import numpy as np

fl\_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1\_9\_left\_seg\_4\_cropped.jpeg'

img=image.load\_img(fl\_img,target\_size=(64,64))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred=np.argmax(model.predict(x))

print(pred)

op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

op[pred]

fl\_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test\_set/A/15.png'

img=image.load\_img(fl\_img,target\_size=(64,64))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred=np.argmax(model.predict(x))

print(pred)

op=['A','B','C','D','E','F','G','H','I']

op[pred]

import tensorflow.keras.models

new\_model = tensorflow.keras.models.load\_model('Hand-SignV2.h5')

fl\_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test\_set/A/15.png'

img=image.load\_img(fl\_img,target\_size=(64,64))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred=np.argmax(new\_model.predict(x))

print(pred)

op=['A','B','C','D','E','F','G','H','I']

op[pred]

fl\_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1\_9\_left\_seg\_4\_cropped.jpeg'

img=image.load\_img(fl\_img,target\_size=(64,64))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred=np.argmax(new\_model.predict(x))

print(pred)

op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

op[pred]

**app.py**

#Import necessary libraries

from flask import Flask, render\_template, Response,url\_for

import cv2

from keras.preprocessing import image

from keras.preprocessing.image import load\_img

import numpy as np

from keras.models import load\_model

#Initialize the Flask app

app = Flask(\_name\_)

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/video\_feed')

def video\_feed():

new\_model = load\_model('Hand-SignV2.h5')

cap = cv2.VideoCapture(0)

try:

while True:

cap.set(cv2.CAP\_PROP\_POS\_MSEC,1000\*1000)

ret, frame = cap.read()

frame = cv2.flip(frame, 1)

cv2.imshow('Input', frame)

if ret:

cv2.imwrite("image.jpeg",frame)

img = image.load\_img("./image.jpeg", target\_size=(64, 64))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

pred = np.argmax(new\_model.predict(x))

op = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

cv2.waitKey(250)

yield op[pred]

except KeyboardInterrupt:

return -1

if \_name=='\_\_main\_':

app.run(debug=True)

**new1.css**

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');

body{

background: #FC7307;

}

a{

color: blue;

text-align: center;

display: block;

font-size:1.8em;

}

.two{

margin-left: auto;

margin-right: auto;

}

h1{

color : #236AB9;

}

h2{

color : #341C09;

}

**7.2) Feature 2**

We used our webcam live feed as input. So when the user shows a hand sign the CNN predicts the correct alphabet or number and prints as a text

The below code is used for this feature

**feed.html**

<html>

<body style="color:#D4E4F7 ">

{{ cv2.capture(0) }}

<br>

{% for i in video\_feed() %}

<h4>{{ i }}</h4>

{% endfor %}

 </body>

</html>

**index.html**

<html>

<head>

<link rel= "stylesheet" type= "text/css" href= "{{ url\_for('static',filename='new1.css') }}">

Video Streaming Live Web Cam

</head>

<title>Real-Time Communication System Powered by AI for Specially Abled</title>

<style>

h1 {text-align: center;}

h2 {text-align: center;}

a {text-align: center;}

</style>

<body>

<div class="container">

<div class="row">

<h1 style="font-size:45px">Real-Time Communication System Powered by AI for Specially Abled</h1>

<h2 style="font-size:35px">Live Streaming</h3>

<a href="{{ url\_for('video\_feed') }}" class="btn btn-warning">Start</a>

</div>

</div>

<img class="one" src="{{ url\_for('static', filename='hand-sign.jpg') }}" width="1000" height="400">

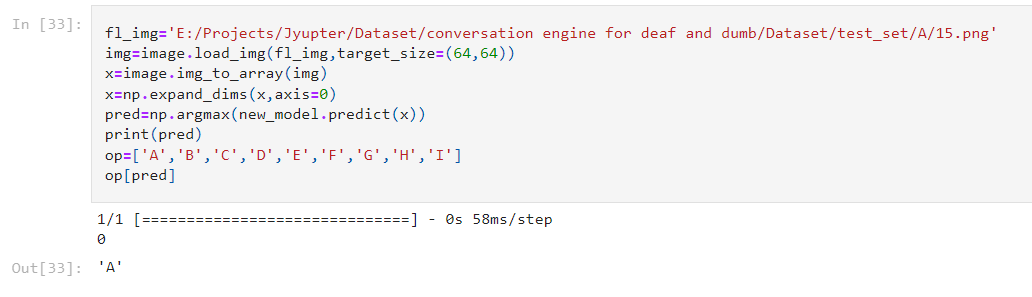
</body>

</html>

**8.) TESTING**

**8.1)TEST CASES**

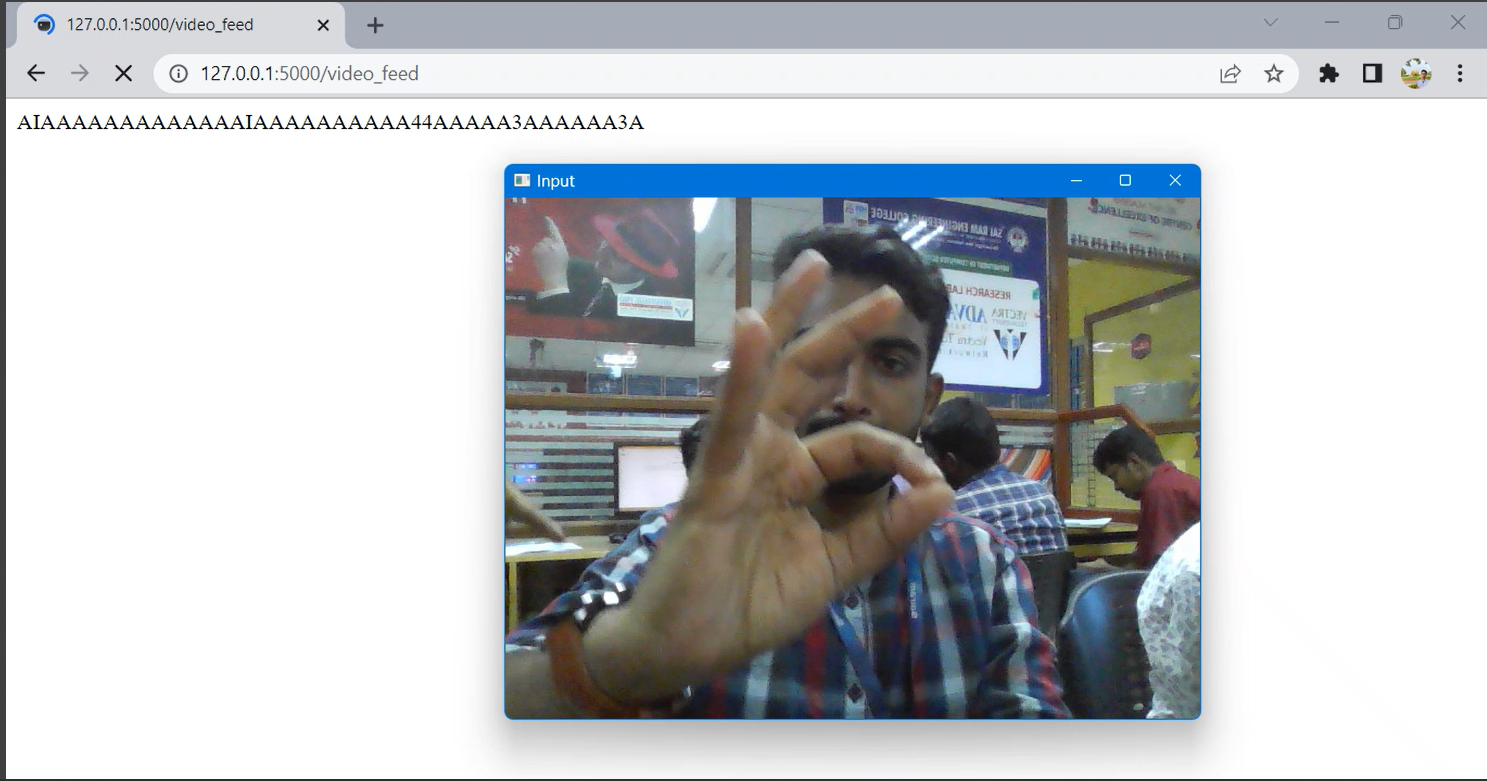
**TEST CASE 1**



**TEST CASE 2**

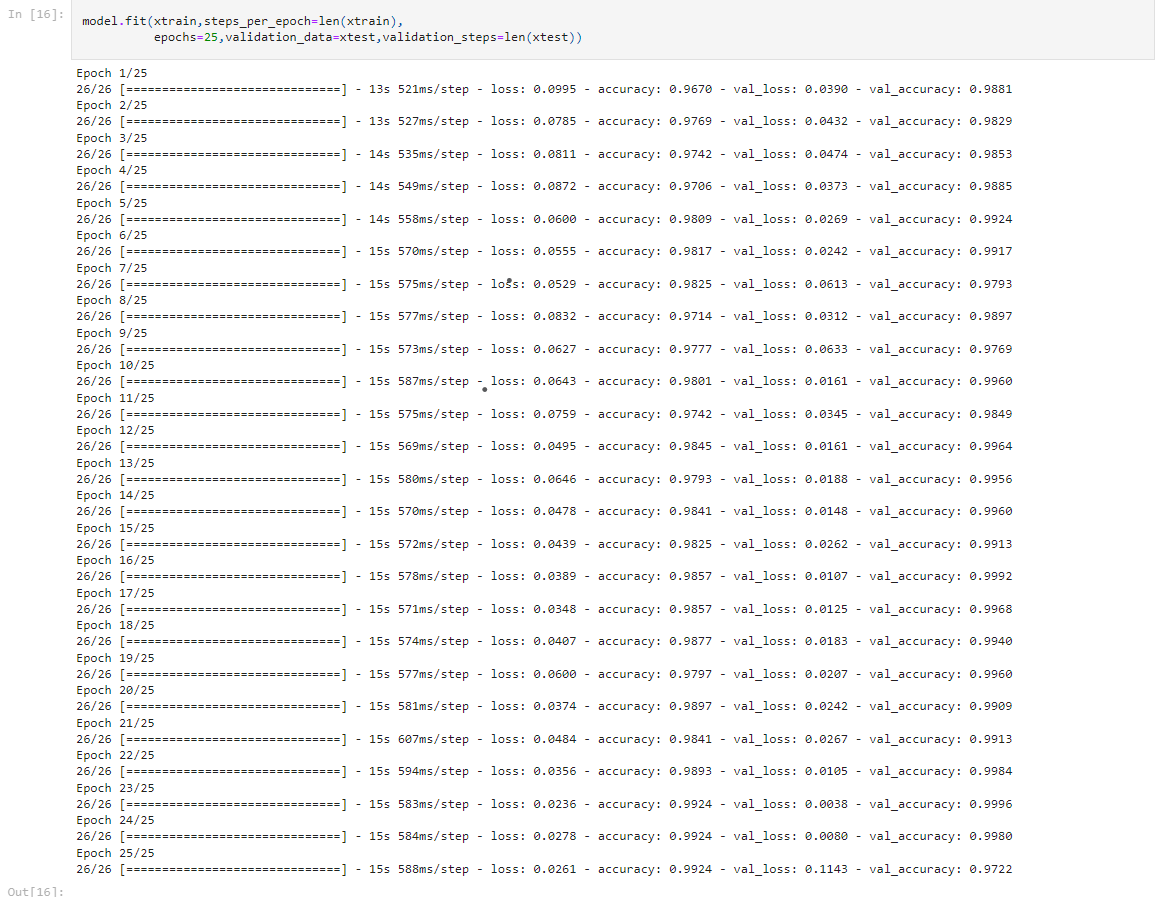


**8.2) USER ACCEPTANCE TESTING**

****

**9) RESULTS**

**9.1) PERFORMANCE METRICS**

****

**10.) ADVANTAGES AND DISADVANTAGES**

**10.1) ADVANTAGES**

\*It is very useful for the deaf and dumb(specially abled) person to communicate with other specially abled person or a normal person using this model.

\*The acuuracy is higher compared to previous models.

\*Before starting the video feed we provide an image which shows what hand sign corresponds to what alphabet , In other words our model is user friendly

**10.2) DISADVANTAGES**

\*Our model is not 100% effective,some flaws in detecting the hand sign may occur.Its mainly because each person use different angles while using hand signs so in detecting the hand sign based on a specific set of data sets is not sufficient.

**11.)CONCLUSION**

The project has developed 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.

**12.)FUTURE SCOPE**

Instead of using only hand signs to communicate. We can introduce basic waving motions of hands , for example waving our hand in horizontal motion means hi etc,

It leads to faster and better way of communication.

Instead of using CNN there can be a chance for a new algorithm in the future which has more accuracy.

The scope for this model in the future is high since there are not many applications for the specially abled persons(dead and dumb) to communicate with one another or with a normal person.

**13.)APPENDIX**

**SOURCE CODE**

**Main code**

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense

train\_data=ImageDataGenerator(rescale=1./255,zoom\_range=0.2,horizontal\_flip=True)

test\_data=ImageDataGenerator(rescale=1./255,validation\_split=0.5)

xtrain=train\_data.flow\_from\_directory('E:/Projects/Jyupter/Dataset/Hand Sign/Train',

target\_size=(64,64),class\_mode='categorical',batch\_size=100)

xtest=test\_data.flow\_from\_directory('E:/Projects/Jyupter/Dataset/Hand Sign/Test',

target\_size=(64,64),class\_mode='categorical',batch\_size=100)

#Layering

model=Sequential()

model.add(Convolution2D(32,(3,3),activation='relu',input\_shape=(64,64,3)))

model.add(MaxPooling2D(pool\_size=(2,2)))

model.add(Flatten())

model.add(Dense(300,activation='relu'))

model.add(Dense(150,activation='relu'))

model.add(Dense(36,activation='softmax'))

#Compile

model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy'])

#Fit

model.fit(xtrain,steps\_per\_epoch=len(xtrain),

epochs=25,validation\_data=xtest,validation\_steps=len(xtest))

model.save('Hand-SignV2.h5')

#Test

from tensorflow.keras.preprocessing import image

import numpy as np

fl\_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1\_9\_left\_seg\_4\_cropped.jpeg'

img=image.load\_img(fl\_img,target\_size=(64,64))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred=np.argmax(model.predict(x))

print(pred)

op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

op[pred]

fl\_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test\_set/A/15.png'

img=image.load\_img(fl\_img,target\_size=(64,64))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred=np.argmax(model.predict(x))

print(pred)

op=['A','B','C','D','E','F','G','H','I']

op[pred]

import tensorflow.keras.models

new\_model = tensorflow.keras.models.load\_model('Hand-SignV2.h5')

fl\_img='E:/Projects/Jyupter/Dataset/conversation engine for deaf and dumb/Dataset/test\_set/A/15.png'

img=image.load\_img(fl\_img,target\_size=(64,64))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred=np.argmax(new\_model.predict(x))

print(pred)

op=['A','B','C','D','E','F','G','H','I']

op[pred]

fl\_img='E:/Projects/Jyupter/Dataset/Hand Sign/Test/9/hand1\_9\_left\_seg\_4\_cropped.jpeg'

img=image.load\_img(fl\_img,target\_size=(64,64))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred=np.argmax(new\_model.predict(x))

print(pred)

op=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

op[pred]

**app.py**

#Import necessary libraries

from flask import Flask, render\_template, Response,url\_for

import cv2

from keras.preprocessing import image

from keras.preprocessing.image import load\_img

import numpy as np

from keras.models import load\_model

#Initialize the Flask app

app = Flask(\_name\_)

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/video\_feed')

def video\_feed():

new\_model = load\_model('Hand-SignV2.h5')

cap = cv2.VideoCapture(0)

try:

while True:

cap.set(cv2.CAP\_PROP\_POS\_MSEC,1000\*1000)

ret, frame = cap.read()

frame = cv2.flip(frame, 1)

cv2.imshow('Input', frame)

if ret:

cv2.imwrite("image.jpeg",frame)

img = image.load\_img("./image.jpeg", target\_size=(64, 64))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

pred = np.argmax(new\_model.predict(x))

op = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

cv2.waitKey(250)

yield op[pred]

except KeyboardInterrupt:

return -1

if \_name=='\_\_main\_':

app.run(debug=True)

**new1.css**

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');

body{

background: #FC7307;

}

a{

color: blue;

text-align: center;

display: block;

font-size:1.8em;

}

.two{

margin-left: auto;

margin-right: auto;

}

h1{

color : #236AB9;

}

h2{

color : #341C09;

}

**feed.html**

<html>

<body style="color:#D4E4F7 ">

{{ cv2.capture(0) }}

<br>

{% for i in video\_feed() %}

<h4>{{ i }}</h4>

{% endfor %}

 </body>

</html>

**index.html**

<html>

<head>

<link rel= "stylesheet" type= "text/css" href= "{{ url\_for('static',filename='new1.css') }}">

Video Streaming Live Web Cam

</head>

<title>Real-Time Communication System Powered by AI for Specially Abled</title>

<style>

h1 {text-align: center;}

h2 {text-align: center;}

a {text-align: center;}

</style>

<body>

<div class="container">

<div class="row">

<h1 style="font-size:45px">Real-Time Communication System Powered by AI for Specially Abled</h1>

<h2 style="font-size:35px">Live Streaming</h3>

<a href="{{ url\_for('video\_feed') }}" class="btn btn-warning">Start</a>

</div>

</div>

<img class="one" src="{{ url\_for('static', filename='hand-sign.jpg') }}" width="1000" height="400">

</body>

</html>

**Project\_Default.xml**

<component name="InspectionProjectProfileManager">

<profile version="1.0">

<option name="myName" value="Project Default" />

<inspection\_tool class="PyPep8Inspection" enabled="true" level="WEAK WARNING" enabled\_by\_default="true">

<option name="ignoredErrors">

<list>

<option value="E302" />

<option value="E305" />

</list>

</option>

</inspection\_tool>

<inspection\_tool class="PyPep8NamingInspection" enabled="true" level="WEAK WARNING" enabled\_by\_default="true">

<option name="ignoredErrors">

<list>

<option value="N801" />

</list>

</option>

</inspection\_tool>

</profile>

</component>

**profiles\_settings.xml**

<component name="InspectionProjectProfileManager">

<settings>

<option name="USE\_PROJECT\_PROFILE" value="false" />

<version value="1.0" />

</settings>

</component>

**IBM.iml**

<?xml version="1.0" encoding="UTF-8"?>

<module type="PYTHON\_MODULE" version="4">

<component name="NewModuleRootManager">

<content url="file://$MODULE\_DIR$" />

<orderEntry type="jdk" jdkName="Python 3.9" jdkType="Python SDK" />

<orderEntry type="sourceFolder" forTests="false" />

</component>

</module>

**misc.xml**

<?xml version="1.0" encoding="UTF-8"?>

<project version="4">

<component name="ProjectRootManager" version="2" project-jdk-name="Python 3.9" project-jdk-type="Python SDK" />

<component name="PyCharmProfessionalAdvertiser">

<option name="shown" value="true" />

</component>

</project>

**modules.xml**

<?xml version="1.0" encoding="UTF-8"?>

<project version="4">

<component name="ProjectModuleManager">

<modules>

<module fileurl=<file://$PROJECT_DIR$/.idea/IBM.iml> filepath="$PROJECT\_DIR$/.idea/IBM.iml" />

</modules>

</component>

</project>

GITHUB LINK:- <https://github.com/IBM-EPBL/IBM-Project-24768-1659948659>

DEMO LINK:- <https://www.youtube.com/watch?v=KwFK2SQf3qM>