# Project Report

## 1. INTRODUCTION

1.1 Project Overview
1.2 Purpose

## 2. LITERATURE SURVEY

2.1 Existing problem
2.2 References
2.3 Problem Statement Definition

## 3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas
3.2 Ideation & Brainstorming
3.3 Proposed Solution
3.4 Problem Solution fit

## 4. REQUIREMENT ANALYSIS

4.1 Functional requirement
4.2 Non-Functional requirements

## 5. PROJECT DESIGN

5.1 Data Flow Diagrams
5.2 Solution & Technical Architecture
5.3 User Stories

## 6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation
6.2 Sprint Delivery Schedule
6.3 Reports from JIRA

## 7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1
7.2 Feature 2
7.3 Database Schema (if Applicable)

Team ID: PNT2022TMID42224

# REAL TIME COMMUNICATION SYSTEM POWERED FOR SPECIALLY ABLED

## 1.INTRODUCTION:

### 1.1 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. 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.

### 1.2 PURPOSE:

Voice-assisted AI technologies such as Alexa, Echo, Siri, and many more are some of the main communication improvements that assist from everyday chores to voice payments. These advancements are mainly targeted to service individuals with visual impairments, boasting their communication with smart devices. Moreover, AI is currently improving cognitive systems to help describe images for people with visual impairments. Furthermore, communication progressions in AI are not only limited to the visually impaired. The development of speech-to-text and text-to-speech technology provides even more interactive opportunities for individuals recovering from brain injuries, such as people who have cerebral palsy. For individuals with hearing impairments, AI offers closed captioning programs—such as the ones found on YouTube—to boost understanding.  The dynamism of AI technologies reaches every industry,

pushing its strength to develop convenience by increasing communication, not just for convenience.

# 2.LITERATURE SURVEY:

## 2.1 EXISTING PROBLEM:

Communication is the only medium by which we can share our thoughts or convey the message but for a person with disability (deaf and dumb) faces difficulty in communication with normal person. Because of this, a person who lacks in hearing and speaking ability is not able to stand in race with normal person. When someone realizes they're interacting with a hearing impaired person, they often switch to a slower form of speech. While it's done with the best intentions, it can actually hinder lip reading. Over time, the hearing impaired have learned to understand words when people speak naturally, so slowing it down intentionally can result in miscommunication.Whether it's a dimly-lit room or a noisy dark club, the absence of light makes it difficult for the hearing impaired to engage with others. They generally rely on visual stimuli, such as lip reading or sign language, so darkness poses a challenge.

## 2.2 REFERENCES:

| S.NO | TITLE | AUTHORS | DESCRIPTION |
|---|---|---|---|
| 1. | Innovative study of an AI voicebased smart device to assist deafpeople | DhayaSindhuBattina | Developmentofanartificialintelligencevoice-based.Smart device that include the Flex sensors, LCD module, microcontroller, SD card memory, hearing.phones,etc.improves the quality of life without the assistance of some artificial means. |

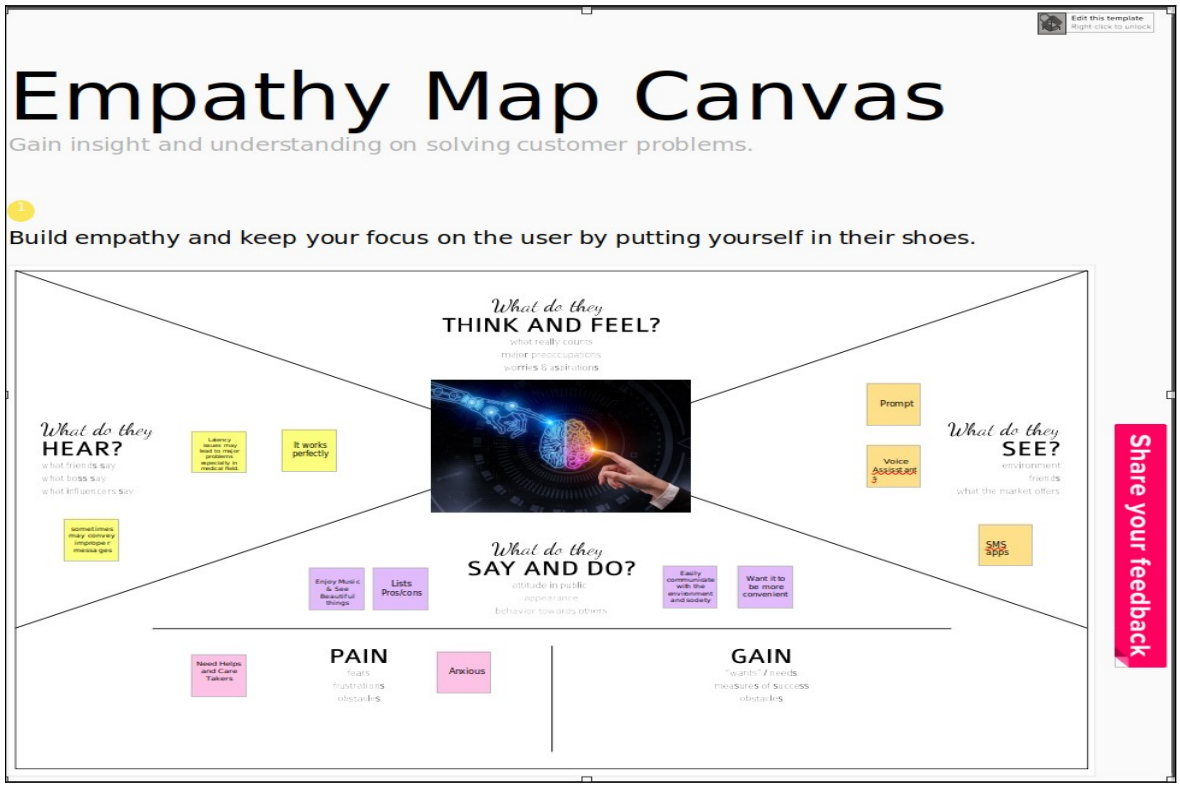
| | | | |
|---|---|---|---|
| 2 | Two Hand Indian SignLanguaged atasetforbench marking classification models of MachineLearni ng | LeelaSuryaTejaMan gamuri,Lakshay JainAbhishekSharma y | This dataset wasbenchmared on six different classification models ofmachine learning bychanging the parameters.Classific ation modelsare evaluated based onthe HOG featuresextracted from the skinfilteredimage.. |
| 3 | Double Handed Indian SignLanguagetoSp eechandText | KusurnikaKroriDut ta,SatheeshKumar Raju,AnilKumarG,S unnyArokiaSwarny | The system is trainedwithdoublehand edsignlanguage by using aminimumeigenvalue algorithm.<br><br>Here Logitech webcameraisusedforim ageacquisition andprocessing is performedinMATLAB. |
| 4 | Indian Sign Language Animation Generation System | Sandeep Kaur,Maninder Singh | This paper describes asystem which generates HamNoSys corresponding to 100words. These Notations are generated according to the Indian Sign Language |

## 2.3 PROBLEM STATEMENT:

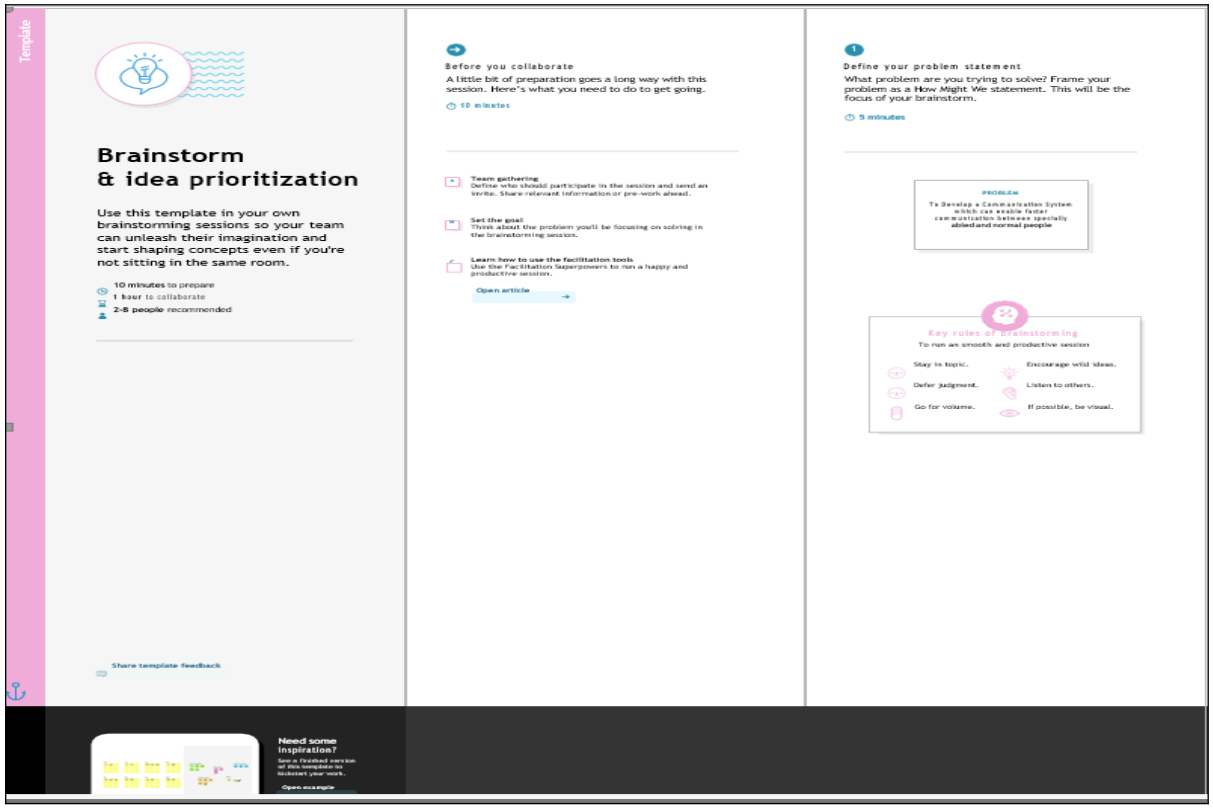
To Develop a communication system which can enable faster communication between specially abled and normal people.The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into 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. The machine is made to learn all the actions so that it can easily detects the movement of the person.Real-time captioning or translations for people with a hearing impairment or even people who don't speak the language.

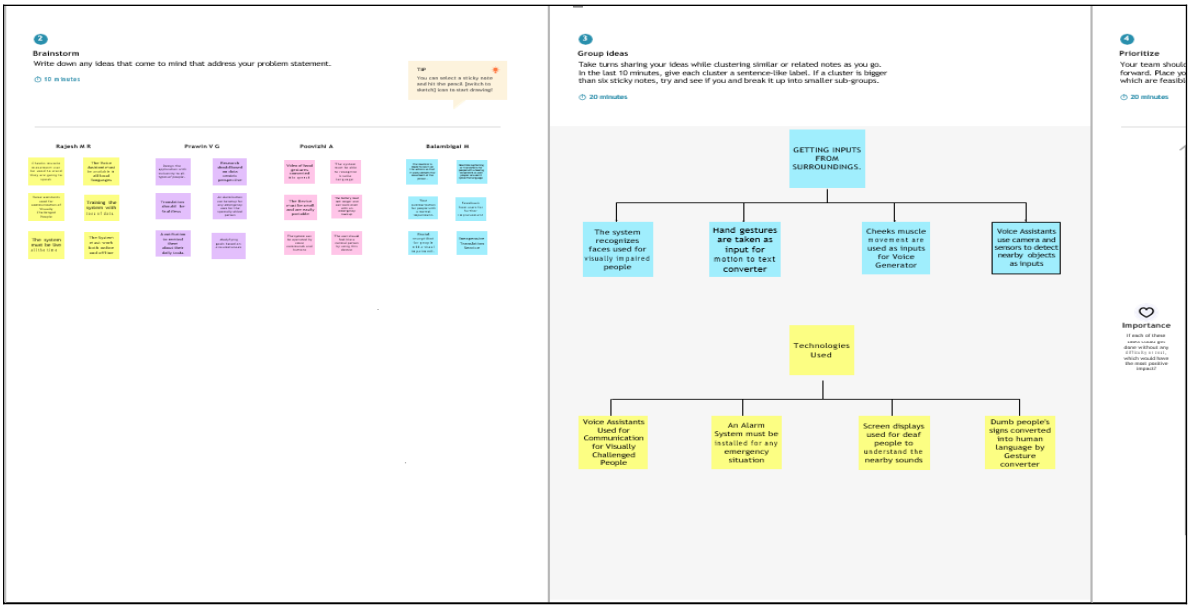
## 3.IDEATION & PROPOSED SOLUTION:

### 3.1 EMPATHY MAP CANVAS:



# 3.2 IDEATION & BRAINSTORMING:



## 3.3 PROPOSED SOLUTION:

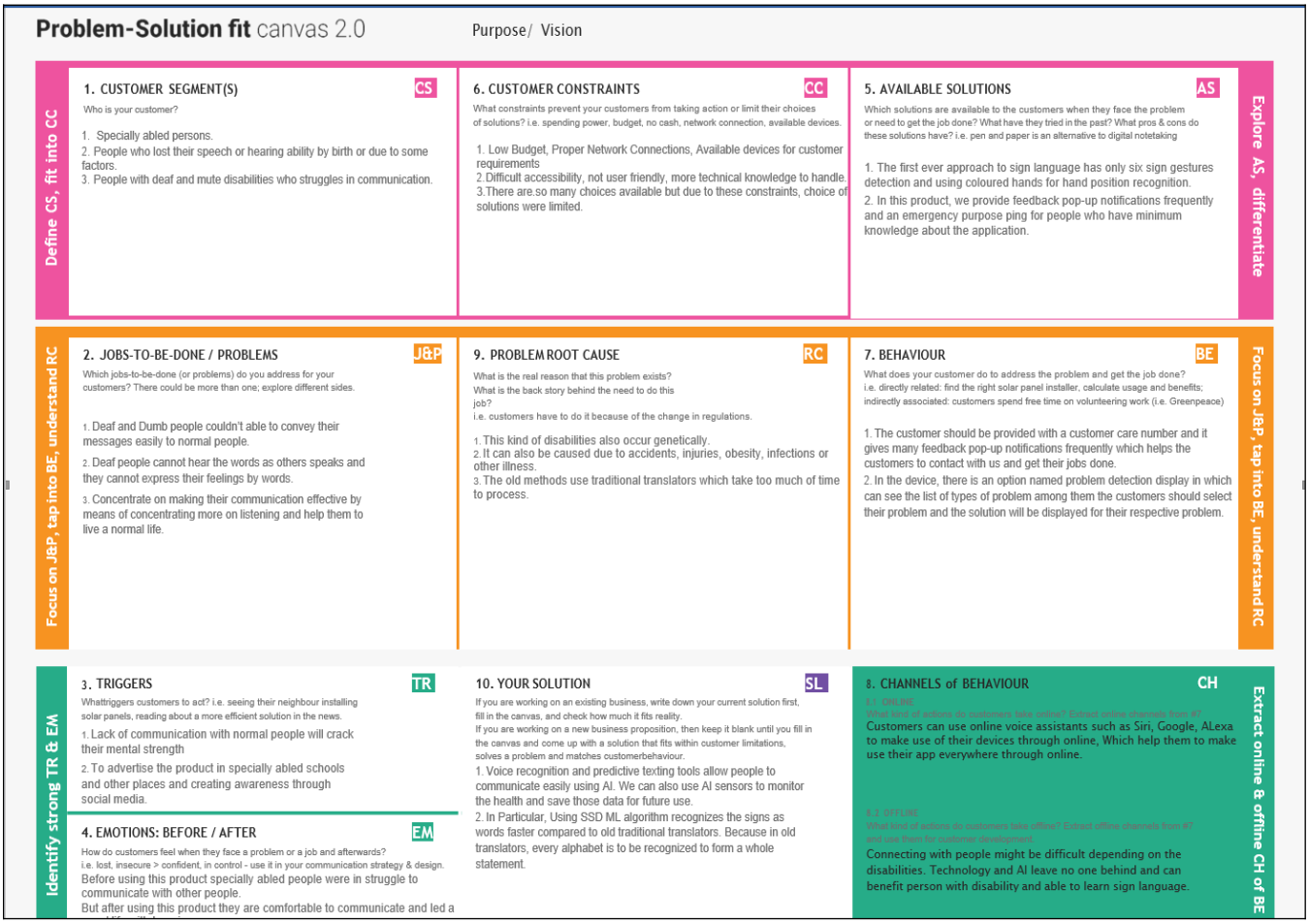
**Proposed Solution Template:**

Project team shall fill the following information in proposed solution template.

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | Communication between deaf-mute and a normal personhas always been a challenging task.It is very difficult for mute people to convey their message to normalpeople in emergency timesas well as innormal times. |
| 2. | Idea/solution description | 1.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.<br><br>2.To Design and Implement a system using artificial intelligence, image processing and data mining concepts to take input as hand gesture. |
| 3. | Novelty/Uniqueness | We are using a convolution neural network to create a model |
| 4. | social impact/customer satisfication | 1. The main purpose of this application is to make deaf-people feel independent andmore confident.<br><br>2. About two thirds of people with a mobilityand dexterity |

| | | |
|---|---|---|
| | | disability are most likely to experience a great deal of difficulty with everyday activities. |
| 5. | Business model | 1. AI generate revenue through direct customers and collaborate with health care sector and generate revenue from their customers.<br><br>2. B2B setting uses to employ deaf and mute employees can use to convey messages according to the company. |
| 6. | Scalability of the solution | 1. Enhance people with disabilities to step into a world where their are facing difficulties in communication.<br><br>2. AI technology hlps disabled person to open up new opportunities for accessibility inclusion in society and independent living.<br><br>3It might open the door to more cutting edge ad creative approaches to the most difficult problems facing disadvantaged people. |

## 3.4 PROBLEM SOLUTION FIT:




## 4.REQUIREMENT ANALYSIS:

### 4.1 FUNCTIONAL REQUIREMENT:

## FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Form<br>Registration through Gmail<br>Registration throughLinkedIN |

| | | |
|---|---|---|
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | User Verification | The user should receive a verification e-mail which they have to confirm to complete the registration. |
| FR-4 | Authentication | The data inside the device is need to be authenticated to ensurethe privacy of the users. |
| FR-5 | Legal Requirements | Proper Medical Certificate is produced to ensure the integrity of the users. |

## 4.2 NON-FUNCTIONAL REQUIREMENT:

## Non-functional Requirements:

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

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | The designed system is easy to use for speciallyabled persons as it is portable andplatform independent. |
| NFR-2 | **Security** | The system should protect the users data in a secure manner and avoid eavesdropping and such activities by means of encryption and decryption |
| NFR-3 | **Reliability** | The system is tested with large number of data in order to maintain the reliability of the users which is needed moston now a days. |
| NFR-4 | **Performance** | The response timeshould be faster thatimproves the performance which is essential to cope up with the challenging world. |
| NFR-5 | **Availability** | The systemis available on 24/7 to use only the internet is needed for effective communication. |
| NFR-6 | **Scalability** | The designed system should needto increase it's performance whenever need is more and response to changesin processing demands. |

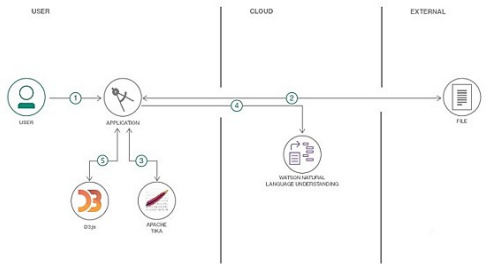
# 5.PROJECT DESIGN:

## 5.1 DATA FLOW DIAGRAMS:

The system should protect the users data in a secure manner and avoid eavesdropping and such activities by means of encryption and decryption.

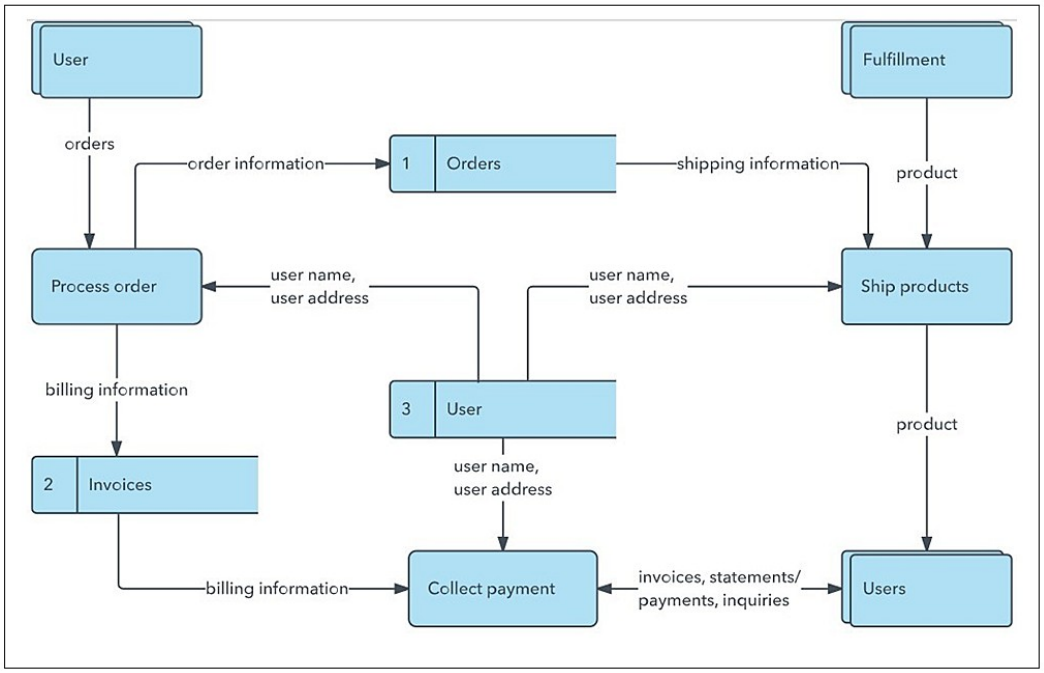
## Example: (Simplified)

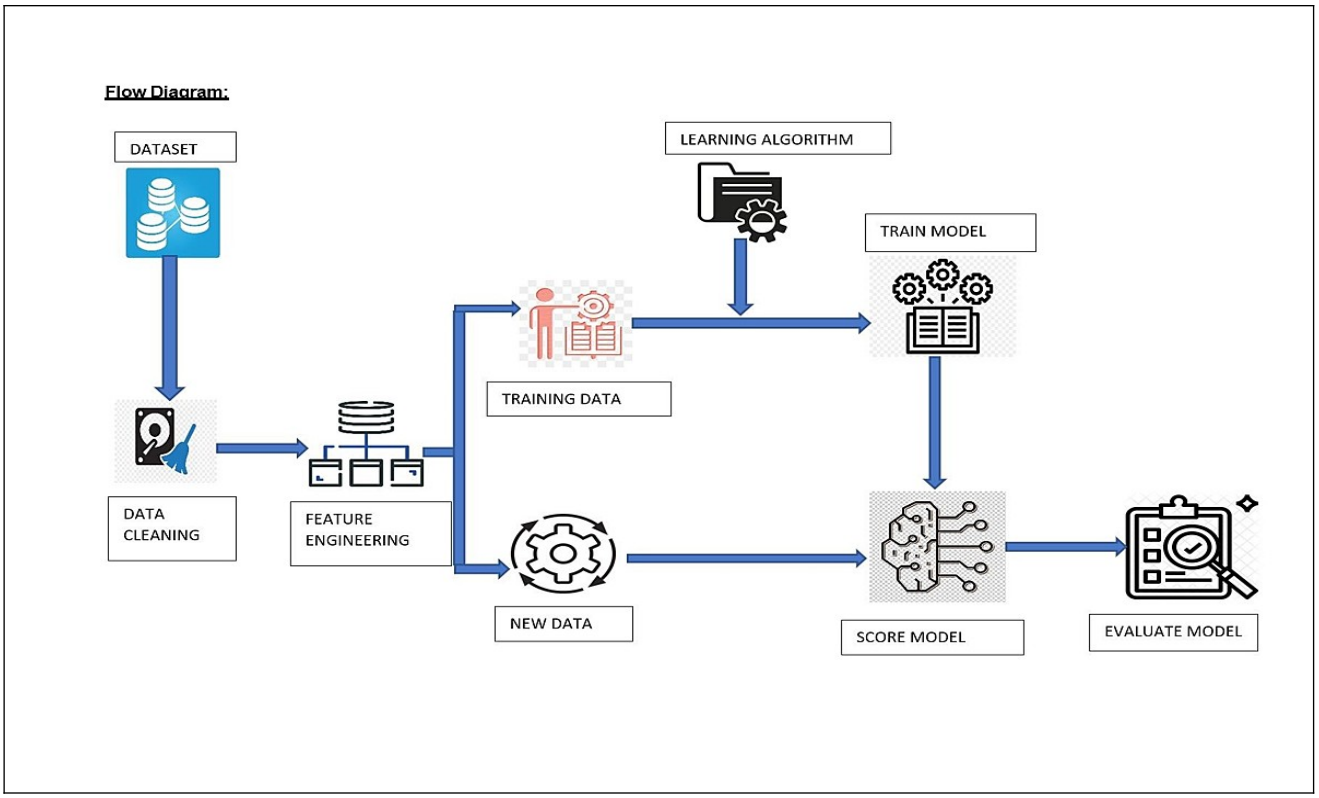
Flow




1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.

## Example: DFD Level 0 (Industry Standard)






## 5.2 SOLUTION & TECHNICAL ARCHITECTURE:

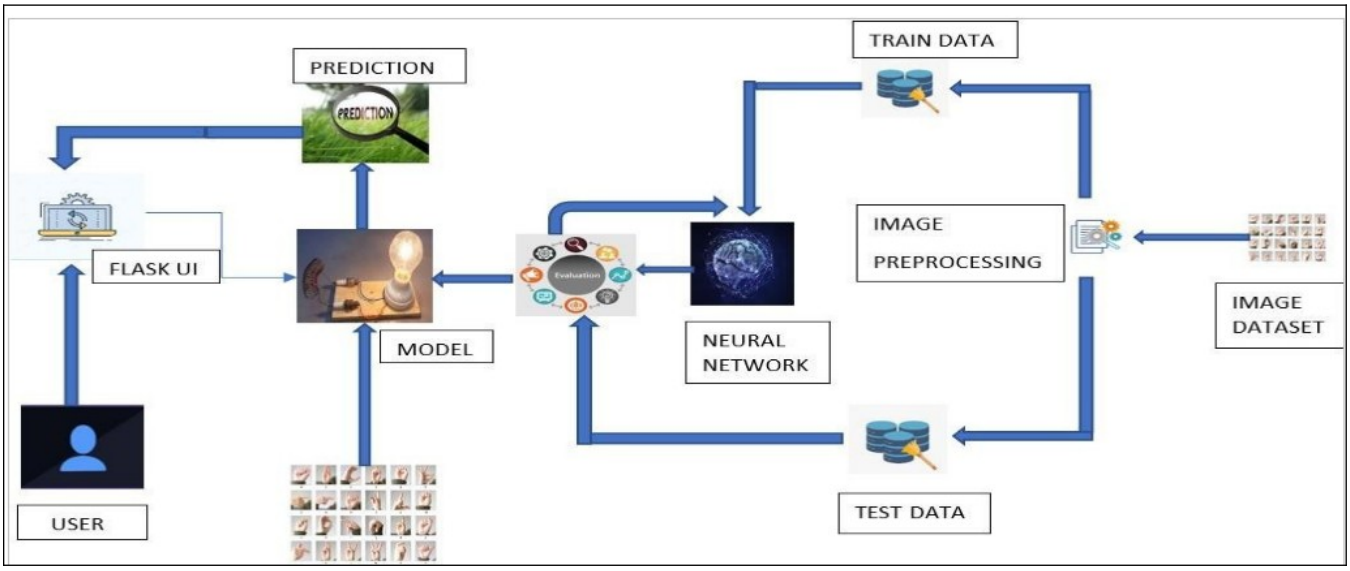



**Table-1 :Components &Technologies:**

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | UserInterface | How user interacts with | HTML,CSS,JavaScript/AngularJs/React Js. |

| | | | |
|---|---|---|---|
| | | application e.g.WebUI,Mob ileApp,Chatbot etc. | |
| 2. | ApplicationLog ic-1 | It deals with variety of frameworks, libraries andsupportsrequired todeveloptheproject | Java/Python |
| 3. | ApplicationLog ic-2 | Helpsinconvertinghuma nvoiceintowritten words, In simple it is used to convert speech totext. | IBMWatsonSTTservice |
| 4. | ApplicationLog ic-3 | Providesfast,consistenta ndaccurateanswers duringtheexecution phaseoftheproject | IBMWatsonAssistant |
| 5. | Database | Itcanbenumerical,catego ricalortime-seriesdata | MySQL,NoSQL,etc. |
| 6. | CloudDatabase | Enables the user to use host database withoutbuyingthead ditionalhardware | IBMDB2, IBMCloudantetc. |
| 7. | FileStorage | File storage should be highly flexible, scalable andeffective | IBMBlockStorageorOtherStorageService |
| 8. | ExternalAPI-1 | Usedtoaccesstheinforma tion inthecloud | IBMWeatherAPI,etc. |
| 9. | ExternalAPI-2 | Used to access the information for data drivendecisionmaki ng | AadharAPI,etc. |
| 10. | MachineLearni ngModel | Machine Learning Model deals with variousalgorithmsthat areneededfortheimplem entation | Real time communication using AI forspeciallyabled |
| 11. | Infrastructure( Server/Cloud) | ApplicationDeploymento nLocalSystem/ CloudLocalServerConfig uration: | Local,CloudFoundry,Kubernetes,etc. |

| | | | |
|---|---|---|---|
| | | Install the windows version and execute theinstaller SelectAPACHE toinstallwebserver | |

**Table-2: ApplicationCharacteristics:**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-SourceFrameworks | Theframeworksusedare | Tensorflow,Theano,RNN,PyTorch,Caffle 2 |
| 2. | SecurityImplementations | the security / access controls implemented, use offirewallsetc. | Identify,PreventazndRespond |
| 3. | ScalableArchitecture | thescalabilityofarchitecture(3–tier,Micro-services) | Data , models, operate at size, speedandcomplexity |
| 4. | Availability | theavailabilityof application(e.g. useofloadbalancers, distributed serversetc.) | Imageandfacialrecognition,lip reading, text summarization, real timecaptioning |
| 5. | Performance | Designconsiderationforth heperformance ofthe application (number of requests per sec, use ofCache,useofCDN’s) etc. | Fullandeffectiveparticipation,equalityof opportunity, accessibility |

## 5.3 USER STORIES:

Use thebelow template to listall theuser stories forthe product.

| UserType | Functional Requirement(Epic) | UserStoryNumber | UserStory/Task | Acceptancecriteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer(Mobileuser) | Registration | USN-1 | Asauser,Icanregisterforthe application byenteringmyemail,password,andconfirming mypassword. | I can access my account /dashboard. | High | Sprint-1 |
| | | USN-2 | Asauser,Iwillreceiveconfirmation emailoncelhaveregistered fortheapplication. | Icanreceiveconfirmationemail& clickconfirm. | High | Sprint-1 |
| | | USN-3 | Asauser,Icanregisterforthe applicationthroughFacebook. | Icanregister&accessthedashboardwithFacebook Login. | Low | Sprint-2 |
| | | USN-4 | Asauser,Icanregisterforthe applicationthroughGmail. | I can register theapplication using gmailwithdetailslinkedtothe gmail. | Medium | Sprint-1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Login | USN-5 | Asauser,Icanlogintotheapplicationbyenteringemail&password. | Can enter thesecredentialseither by manualor by autofilling depends onthe case. | High | Sprint-1 |

## 6. PROJECT PLANNING & SCHEDULING:

### 6.1 SPRINT PLANNING AND ESTIMATION:

| **Sprint** | **FunctionalRequirement(Epic)** | **UserStoryNumber** | **UserStory/Task** | **StoryPoints** | **Priority** | **TeamMembers** |
|---|---|---|---|---|---|---|
| Sprint-1 | Registration | USN-1 | Asauser,Icanregisterfortheapplicationbyentering my email, password, and confirmingmypassword. | 5 | High | Rajesh M R<br>Prawin V G<br>Poovizhi A<br>BalambigaiM |
| Sprint-1 | | USN-2 | Asauser,Iwillreceiveconfirmationemailonce Ihaveregisteredfortheapplication. | 5 | High | Rajesh M R<br>Prawin V G<br>Poovizhi A<br>BalambigaiM |
| Sprint-1 | | USN-3 | Asauser,Icanregi | 5 | Mediu | Rajesh M R |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | sterfortheapplicationthroughGmail. | | m | Prawin V G<br>Poovizhi A<br>BalambigaiM |
| Sprint-1 | Login | USN-4 | Asauser,Icanlogintotheapplicationbyenteringemail&password. | 5 | High | Rajesh M R<br>Prawin V G<br>Poovizhi A<br>BalambigaiM |
| Sprint-2 | DataCollection | USN-5 | CollectingtheRequiredDataset. | 10 | High | Rajesh M R<br>Prawin V G<br>Poovizhi A<br>BalambigaiM |

| **Sprint** | **Functional Requirement (Epic)** | **User StoryNumber** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
|---|---|---|---|---|---|---|
| Sprint-2 | Data cleaning andImage Preprocessing | USN-6 | Perform the imagepreprocessing techniques on the dataset. | 10 | High | Rajesh M R<br>Prawin V G<br>Poovizhi A<br>BalambigaiM |
| Sprint-3 | Model Building | USN-7 | Model Initialization with required layers. | 10 | High | Rajesh M R<br>Prawin V G<br>Poovizhi A<br>BalambigaiM |
| Sprint-3 | Training | USN-8 | Training the image classification model | 10 | High | RajeshM R<br>Prawin V G<br>Poovizhi A<br>Balambigai M |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | using theNeural Network. | | | |
| Sprint-4 | Testing | USN-9 | Testing the Model's Performance. | 10 | High | RajeshM R Prawin V G Poovizhi A Balambigai M |
| Sprint-4 | Deployment of model in Web / App | USN-10 | Deploying the Tested Model | 10 | Medium | Rajesh M R Prawin V G Poovizhi A Balambigai M |

## 6.2 SPRINT DELIVERY SCHEDULE:

| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned EndDate)** | **Sprint Release Date(Actual)** |
|---|---|---|---|---|---|---|
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 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 | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 NOV 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 NOV 2022 |

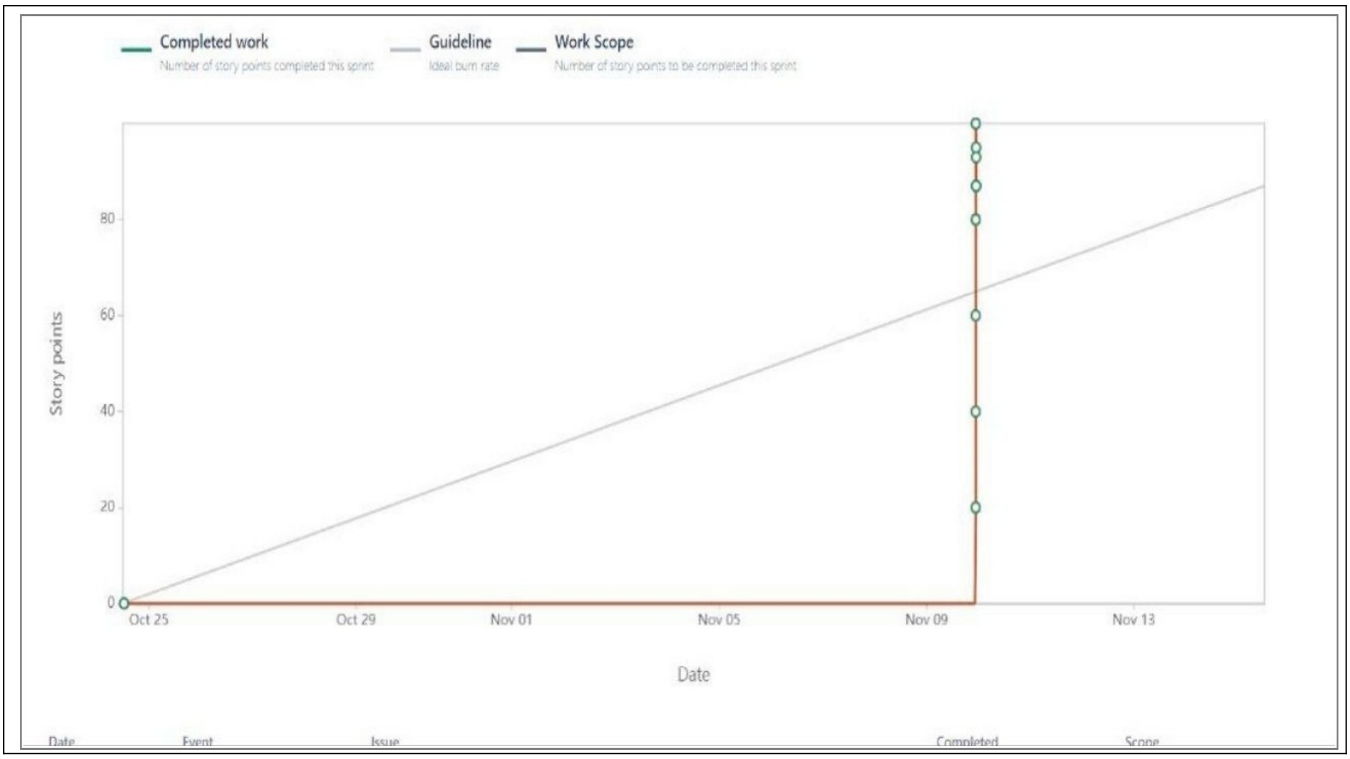
## 6.3 REPORTS FROM JIRA:

Velocity:
Imagine we have a 10-day sprint duration, and the velocityof the team is 20 (points per sprint). Let’s calculate the team’s average velocity(AV) per iterationunit (story points per day)

AV = sprint duration / velocity
= 20 / 10
= 2

### Burndown Chart:

A burn downchart is a graphical representation of work left to do versus time. Itis often used in agile software development methodologies such as Scrum. However, burn down chartscan be applied to any project containing measurable progressover time.




## 7. CODING & SOLUTIONING:

INDEX.HTML:

```
<!DOCTYPE html>
  <html>
   <head>
      <link rel="stylesheet" type="text/css" href="style.css">
      <link rel= "stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">
      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css"
      integrity="sha512-Fo3rlrZj/k7ujTnHg4CGR2D7kSs0v4LLanw2qksYuRlEzO+tcaEPQogQ0KaoGN26/zrn20ImR1DfuLWnOo7aBA==" crossorigin="anonymous" referrerpolicy="no-referrer" />
      <title>Real Time Communication</title>

   <body>
      <!-- <img src="./img1.jpg" style="width:100%"  class="image1" alt="bodyimg"> -->
    <div class="title">
      <h1>
         REAL-TIME COMMUNICATION SYSTEM POWERED BY AI
         <BR>FOR SPECIALLY ABLED
      </h1>
    </div>
     <div>
        <p>
           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.
        </p>
     </div>
      <div class="right">
          <h2 style="align-content: center">TO OPEN THE CAMERA FOR RECOGNITION</h2>
          <button class="button1"><a href="{{ url_for('predict') }}">CLICK HERE</a></button>
      </div>



   </body>
</html>
```

# STYLE.CSS:

```
body{
  /* background-image: linear-gradient(to top,rgba(0,0,0,0.3) ,rgba(0, 0, 0, 0.6)),url('./img1.jpg'); */
  background-image: url(./img2.webp);
/*
    background-size:auto; */
    background-size: contain;
    background-position:top;
    background-repeat: no-repeat;
    margin: 0;
   height: 500vh;
    overflow:hidden;
        padding:0;
    text-align: center;
     width: 100%;
}

.myvideo {
    background-color: floralwhite;
    margin-top: 10vh;
    align-content: center;
}
.title{
    text-align: center;
    color: white;
    margin-top: 10vh;

}

h2{
    color: white;
    text-decoration: none;
    font-size: 30px;
    text-align: right;
    padding-right: 120px;

}
a{
    animation:animate 1s ease-in-out infinite alternate;
    color: #c9e1f5;
    font-size: 25px;
    font-style: unset;
    text-decoration: none;
    text-align: right;
    /* align-items: right; */

}
```

```
.right{
    align-items: right;
}
.button1{
    background-color: steelblue;
    text-align: right;
    margin-left: 990px;
    margin-top: 20px;
}
a:hover{
    transform: scale(1.1);
    transition: transform 0.3s;
    transition-duration: transform 0.2s;
    text-decoration: underline;
    padding-bottom: 10px;
}
p{
    color: #f0f8ff;
    align-content: center;
    margin: 7vw;
    font-size: 30px
}
@keyframes animate {
    from{
        text-shadow: 0 0 30px #141414;
    }
    to{
        text-shadow: 0 0 20px #141414,
                     0 0 10px #141414;
}
}
```

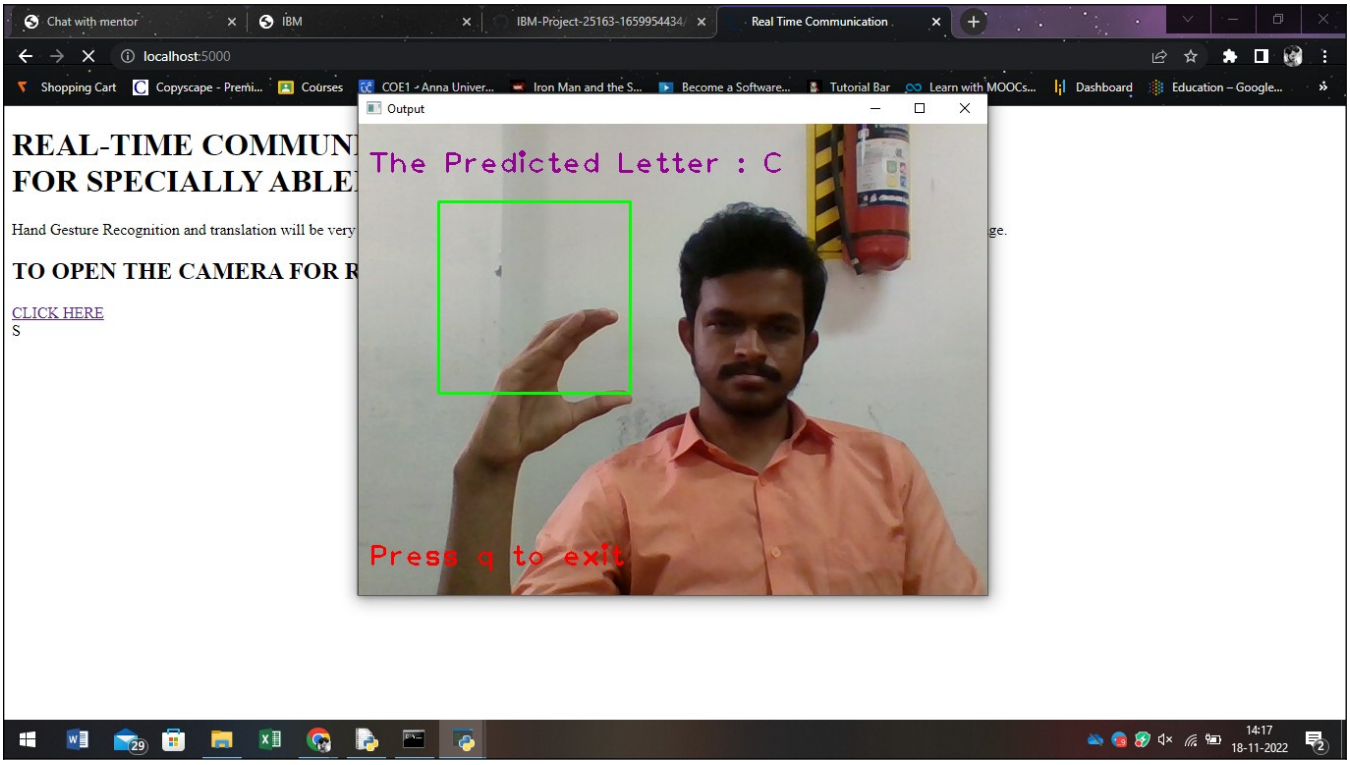
main.py

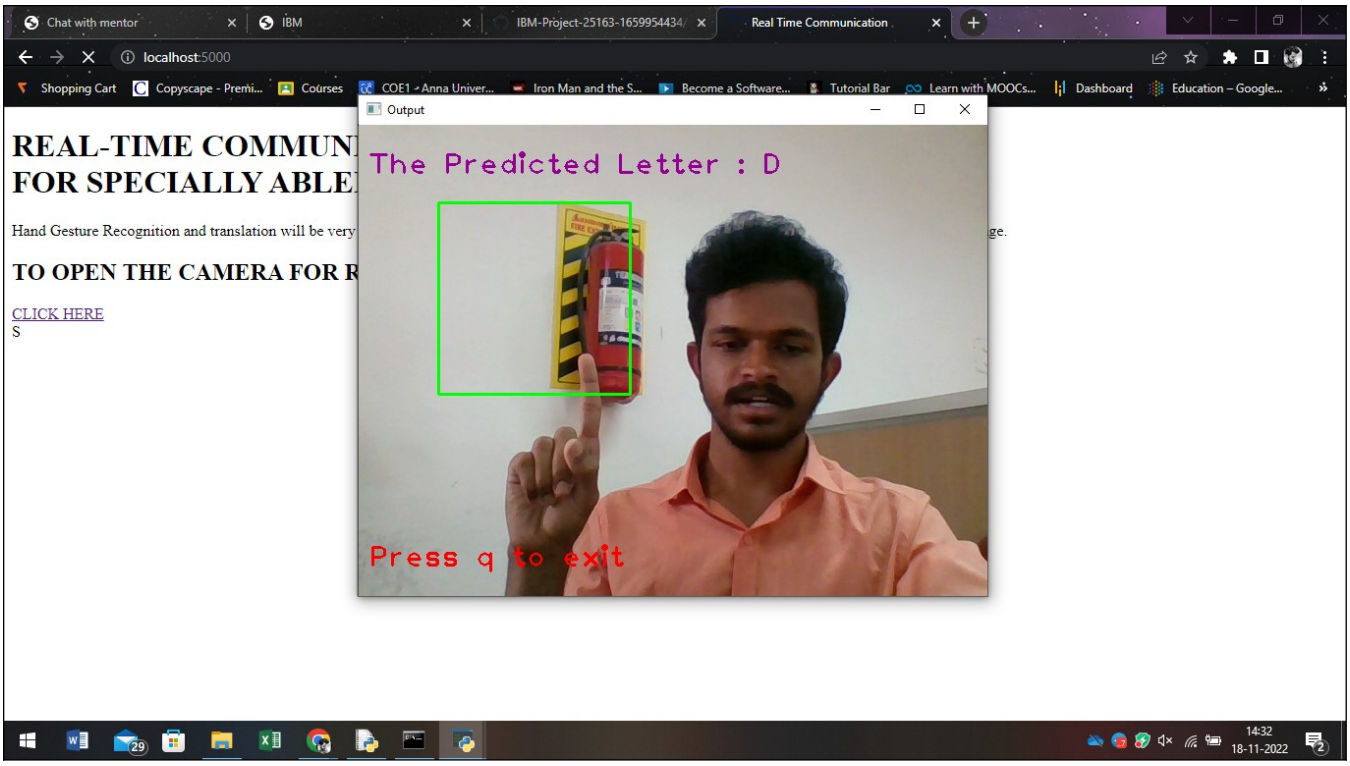
```
from flask import Flask,render_template,request
import cv2
from keras.models import load_model
import numpy as np
from gtts import gTTS
import os
from keras.preprocessing import image
from skimage.transform import resize
app = Flask(__name__)

model=load_model("model.h5")

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

@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')
@app.route('/index', methods=['GET'])
def home():
    return render_template('index.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
        print("[INFO] starting video stream...")
        vs = cv2.VideoCapture(0)

        (W, H) = (None, None)

        while True:
            (grabbed, frame) = vs.read()

            if not grabbed:
                break
```

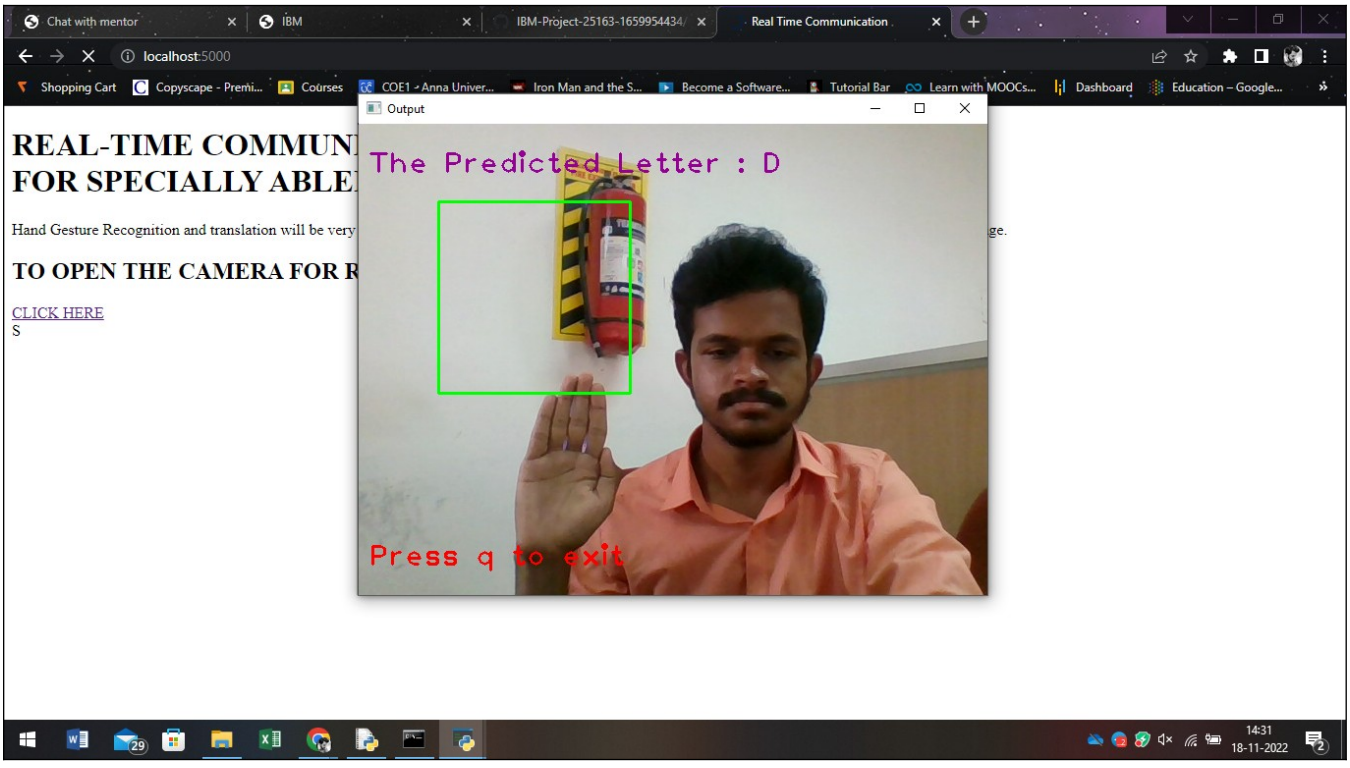
```
            if W is None or H is None:
                (H, W) = frame.shape[:2]
            output = frame.copy()
            r = cv2.selectROI("Select", output)
            print(r)
            cv2.rectangle(output, (81, 79), (276,274), (0,255,0), 2)
            frame = frame[81:276, 79:274]
            frame = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
            _, frame = cv2.threshold(frame, 95, 255, cv2.THRESH_BINARY_INV)
            frame = cv2.cvtColor(frame, cv2.COLOR_GRAY2RGB)


            img = resize(frame,(64,64,3))
            img = np.expand_dims(img,axis=0)
            if(np.max(img)>1):
                img = img/255.0


            result = np.argmax(model.predict(img))
            index=['A', 'B','C','D','E','F','G','H','I']
            result=str(index[result])


            cv2.putText(output, "The Predicted Letter : {}".format(result), (10, 50), cv2.FONT_HERSHEY_PLAIN,
                        2, (150,0,150), 2)
            cv2.putText(output, "Press q to exit", (10,450), cv2.FONT_HERSHEY_PLAIN, 2, (0,0,255), 2)


            speech = gTTS(text = result, lang = 'en', slow = False)

            cv2.imshow("Output", output)
            key = cv2.waitKey(1) & 0xFF
```

```
            if key == ord("q"):
                break


        print("[INFO] cleaning up...")
        vs.release()
        cv2.destroyAllWindows()
        return render_template("index.html")


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

## 8. Output:






## 8.2 Train CV Model on IBM

```
pwd
```

Out[7]:

```
'/home/wsuser/work'
```

In [8]:

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2,
zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes
your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='oIbr0fBehUOMTXjyGQ7PuqVqDDIHvLGdxRmHf9bz_Rxr',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'communicationdeadanddumb-donotdelete-pr-wi3aitewfbnvl0'
object_key = 'conversation engine for deaf and dumb.zip'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']
if not hasattr(streaming_body_1, "__iter__"):streaming_body_1.__iter__ =
types.MethodType(__iter__,streaming_body_1)

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the
possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

```
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

In [12]:

```
import os
filenames = os.listdir('/home/wsuser/work/conversation engine for deaf and
dumb/training_set')
```

In [13]:

```
X_train = train_datagen.flow_from_directory('/home/wsuser/work/conversation
engine for deaf and dumb/training_set', target_size = (64,64), batch_size =
300, class_mode = 'categorical', color_mode = 'grayscale')
X_test = test_datagen.flow_from_directory('/home/wsuser/work/conversation
engine for deaf and dumb/test_set', target_size = (64,64), batch_size = 300,
class_mode = 'categorical', color_mode = 'grayscale')
Found 15750 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
X_train.class_indices
```

Out[14]:

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

In [15]:

```
model = Sequential()
model.add(Convolution2D(32,(3,3), input_shape = (64,64,1), activation =
'relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Flatten())
model.add(Dense(units = 512, activation = 'relu'))
model.add(Dense(units = 256, activation = 'relu'))
model.add(Dense(units = 128, activation = 'relu'))
model.add(Dense(units = 64, activation = 'relu'))
model.add(Dense(units = 9, activation = 'softmax'))
```

In [16]:

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

In [17]:

```
model.fit(X_train, steps_per_epoch = len(X_train),epochs=10, validation_data
= X_test, validation_steps= len(X_test))
Epoch 1/10
53/53 [==============================] - 146s 3s/step - loss: 0.7353 -
accuracy: 0.7479 - val_loss: 0.3811 - val_accuracy: 0.9191
Epoch 2/10
53/53 [==============================] - 142s 3s/step - loss: 0.1022 -
accuracy: 0.9708 - val_loss: 0.1659 - val_accuracy: 0.9653
Epoch 3/10
53/53 [==============================] - 144s 3s/step - loss: 0.0437 -
accuracy: 0.9877 - val_loss: 0.1562 - val_accuracy: 0.9764
Epoch 4/10
53/53 [==============================] - 138s 3s/step - loss: 0.0235 -
accuracy: 0.9937 - val_loss: 0.1719 - val_accuracy: 0.9769
```

```
Epoch 5/10
53/53 [==============================] - 136s 3s/step - loss: 0.0155 -
accuracy: 0.9957 - val_loss: 0.1818 - val_accuracy: 0.9773
Epoch 6/10
53/53 [==============================] - 139s 3s/step - loss: 0.0116 -
accuracy: 0.9969 - val_loss: 0.1925 - val_accuracy: 0.9778
Epoch 7/10
53/53 [==============================] - 139s 3s/step - loss: 0.0073 -
accuracy: 0.9980 - val_loss: 0.1998 - val_accuracy: 0.9693
Epoch 8/10
53/53 [==============================] - 141s 3s/step - loss: 0.0039 -
accuracy: 0.9989 - val_loss: 0.2185 - val_accuracy: 0.9769
Epoch 9/10
53/53 [==============================] - 140s 3s/step - loss: 0.0052 -
accuracy: 0.9983 - val_loss: 0.1956 - val_accuracy: 0.9769
Epoch 10/10
53/53 [==============================] - 141s 3s/step - loss: 0.0030 -
accuracy: 0.9996 - val_loss: 0.2857 - val_accuracy: 0.9773
```

Out[17]:

In [18]:

```
model.save('conv.h5')
```

In [19]:

```
!tar -zcvf conversation_engine.tgz conv.h5
conv.h5
```

In [20]:

```
ls -1
'conversation engine for deaf and dumb'/
conversation_engine.tgz
conv.h5
!pip install watson-machine-learning-client --upgrade
Collecting watson-machine-learning-client
  Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538
kB)
     |████████████████████████████████| 538 kB 12.7 MB/s eta 0:00:01
Requirement already satisfied: certifi in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
learning-client) (2022.9.24)
Requirement already satisfied: tabulate in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
learning-client) (0.8.9)
Requirement already satisfied: urllib3 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
learning-client) (1.26.7)
Requirement already satisfied: ibm-cos-sdk in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
learning-client) (2.11.0)
Requirement already satisfied: lomond in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
learning-client) (0.3.3)
Requirement already satisfied: pandas in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
```

```
learning-client) (1.3.4)
Requirement already satisfied: tqdm in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
learning-client) (4.62.3)
Requirement already satisfied: requests in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
learning-client) (2.26.0)
Requirement already satisfied: boto3 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
learning-client) (1.18.21)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-
machine-learning-client) (0.10.0)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-
machine-learning-client) (0.5.0)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-
machine-learning-client) (1.21.41)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-
dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson-machine-
learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-
>watson-machine-learning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-
>watson-machine-learning-client) (2.11.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-
>watson-machine-learning-client) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-
>watson-machine-learning-client) (3.3)
Requirement already satisfied: pytz>=2017.3 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-
machine-learning-client) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-
machine-learning-client) (1.20.3)
Installing collected packages: watson-machine-learning-client
Successfully installed watson-machine-learning-client-1.0.391
```

In [23]:

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url" : "https://us-south.ml.cloud.ibm.com",
    "apikey" : "e9o1OIOfeMwU3gAmqbrwN2bLpGYDJ7ZdLCgcaz0biAof"
```

```
}
client = APIClient(wml_credentials)
```

In [25]:

```
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if item['entity']
['name'] == space_name)['metadata']['id'])
```

In [26]:

```
space_uid = guid_from_space_name(client, 'conv_engine')
print("Space UID = " + space_uid)
Space UID = 1233b3f7-5150-422d-a41b-523e665492e0
```

In [28]:

```
client.set.default_space(space_uid)
```

Out[28]:

```
'SUCCESS'
```

In [41]:

```
!pip install keras==2.2.4
Collecting keras==2.2.4
  Downloading Keras-2.2.4-py2.py3-none-any.whl (312 kB)
     |████████████████████████████████| 312 kB 13.6 MB/s eta 0:00:01
Requirement already satisfied: six>=1.9.0 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4)
(1.15.0)
Requirement already satisfied: scipy>=0.14 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4)
(1.7.3)
Requirement already satisfied: keras-preprocessing>=1.0.5 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4)
(1.1.2)
Requirement already satisfied: pyyaml in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4)
(5.4.1)
Requirement already satisfied: h5py in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4)
(3.1.0)
Collecting keras-applications>=1.0.6
  Downloading Keras_Applications-1.0.8-py3-none-any.whl (50 kB)
     |████████████████████████████████| 50 kB 9.6 MB/s  eta 0:00:01
Requirement already satisfied: numpy>=1.9.1 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4)
(1.19.5)
Installing collected packages: keras-applications, keras
  Attempting uninstall: keras
    Found existing installation: Keras 2.4.0
    Uninstalling Keras-2.4.0:
      Successfully uninstalled Keras-2.4.0
Successfully installed keras-2.2.4 keras-applications-1.0.8
```

In [42]:

```
client.software_specifications.list(limit =100)
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
```

```
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10         125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                        295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8             2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6           2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9           2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6               2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                 2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6              39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt      40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                    493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6           49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6              4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6          50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8             52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11      55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                  5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1               5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                      5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
runtime-22.2-py3.10-xc          5e8cddff-db4a-5a6a-b8aa-2d4af9864dab  base
```

```
autoai-kb_3.1-py3.7               632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8            634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
spark-mllib_2.3-r_3.6             6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c  base
tensorflow_2.4-py3.7              65e171d7-72d1-55d9-8ebb-f813d620c9bb  base
spss-modeler_18.2                 687eddc9-028a-4117-b9dd-e57b36f1efa5  base
pytorch-onnx_1.2-py3.6            692a6a4d-2c4d-45ff-a1ed-b167ee55469a  base
spark-mllib_2.3-scala_2.11        7963efe5-bbec-417e-92cf-0574e21b4e8d  base
spark-mllib_2.4-py37              7abc992b-b685-532b-a122-a396a3cdbaab  base
caffe_1.0-py3.6                   7bb3dbe2-da6e-4145-918d-b6d84aa93b6b  base
pytorch-onnx_1.7-py3.7            812c6631-42b7-5613-982b-02098e6c909c  base
cuda-py3.6                        82c79ece-4d12-40e6-8787-a7b9e0f62770  base
tensorflow_1.15-py3.6-horovod     8964680e-d5e4-5bb8-919b-8342c6c0dfd8  base
hybrid_0.1                        8c1a58c6-62b5-4dc4-987a-df751c2756b6  base
pytorch-onnx_1.3-py3.7            8d5d8a87-a912-54cf-81ec-3914adaa988d  base
caffe-ibm_1.0-py3.6               8d863266-7927-4d1e-97d7-56a7f4c0a19b  base
runtime-22.2-py3.10-cuda          8ef391e4-ef58-5d46-b078-a82c211c1058  base
spss-modeler_17.1                 902d0051-84bd-4af6-ab6b-8f6aa6fdeabb  base
do_12.10                          9100fd72-8159-4eb9-8a0b-a87e12eefa36  base
do_py3.7                          9447fa8b-2051-4d24-9eef-5acb0e3c59f8  base
spark-mllib_3.0-r_3.6             94bb6052-c837-589d-83f1-f4142f219e32  base
cuda-py3.7-opence                 94e9652b-7f2d-59d5-ba5a-23a414ea488f  base
nlp-py3.8                         96e60351-99d4-5a1c-9cc0-473ac1b5a864  base
cuda-py3.7                        9a44990c-1aa1-4c7d-baf8-c4099011741c  base
hybrid_0.2                        9b3f9040-9cee-4ead-8d7a-780600f542f7  base
spark-mllib_3.0-py38              9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418  base
autoai-kb_3.3-py3.7               a545cca3-02df-5c61-9e88-998b09dc79af  base
spark-mllib_3.0-py39              a6082a27-5acc-5163-b02c-6b96916eb5e0  base
runtime-22.1-py3.9-do             a7e7dbf1-1d03-5544-994d-e5ec845ce99a  base
default_py3.8                     ab9e1b80-f2ce-592c-a7d2-4f2344f77194  base
tensorflow_rt22.1-py3.9           acd9c798-6974-5d2f-a657-ce06e986df4d  base
kernel-spark3.2-py3.9             ad7033ee-794e-58cf-812e-a95f4b64b207  base
autoai-obm_2.0 with Spark 3.0     af10f35f-69fa-5d66-9bf5-acb58434263a  base
runtime-22.2-py3.10               b56101f1-309d-549b-a849-eaa63f77b2fb  base
default_py3.7_opence              c2057dd4-f42c-5f77-a02f-72bdbd3282c9  base
tensorflow_2.1-py3.7              c4032338-2a40-500a-beef-b01ab2667e27  base
do_py3.7_opence                   cc8f8976-b74a-551a-bb66-6377f8d865b4  base
spark-mllib_3.3                   d11f2434-4fc7-58b7-8a62-755da64fdaf8  base
autoai-kb_3.0-py3.6               d139f196-e04b-5d8b-9140-9a10ca1fa91a  base
spark-mllib_3.0-py36              d82546d5-dd78-5fbb-9131-2ec309bc56ed  base
autoai-kb_3.4-py3.8               da9b39c3-758c-5a4f-9cfd-457dd4d8c395  base
kernel-spark3.2-r3.6              db2fe4d6-d641-5d05-9972-73c654c60e0a  base
autoai-kb_rt22.1-py3.9            db6afe93-665f-5910-b117-d879897404d9  base
tensorflow_rt22.1-py3.9-horovod   dda170cc-ca67-5da7-9b7a-cf84c6987fae  base
autoai-ts_1.0-py3.7               deef04f0-0c42-5147-9711-89f9904299db  base
tensorflow_2.1-py3.7-horovod      e384fce5-fdd1-53f8-bc71-11326c9c635f  base
default_py3.7                     e4429883-c883-42b6-87a8-f419d64088cd  base
do_22.1                           e51999ba-6452-5f1f-8287-17228b88b652  base
autoai-obm_3.2                    eae86aab-da30-5229-a6a6-1d0d4e368983  base
```

```
runtime-22.2-r4.2               ec0a3d28-08f7-556c-9674-ca7c2dba30bd  base
tensorflow_rt22.2-py3.10        f65bd165-f057-55de-b5cb-f97cf2c0f393  base
do_20.1                         f686cdd9-7904-5f9d-a732-01b0d6b10dc5  base
------------------------------  ------------------------------------  ----
```

In [33]:

```
import keras
```

In [34]:

```
keras.__version__
```

Out[34]:

```
'2.7.0'
```

In [30]:

```
software_spec_uid =
client.software_specifications.get_uid_by_name("tensorflow_2.4-py3.8-
horovod")
software_spec_uid
```

Out[30]:

```
'217c16f6-178f-56bf-824a-b19f20564c49'
```

In [39]:

```
model_details = client.repository.store_model(model =
'conversation_engine.tgz',meta_props = {
client.repository.ModelMetaNames.NAME:"CV",
client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid}
                                                )
model_id = client.repository.get_model_id(model_details)
```

In [67]:

```
model_id
```

Out[67]:

```
'740d4c73-8c0e-417c-9401-a50be4747d12'
```

In [68]:

```
client.repository.download(model_id,'my_model.tar.gz')
Successfully saved model content to file: 'my_model.tar.gz'
```

Out[68]:

```
'/home/wsuser/work/my_model.tar.gz'
```

In [69]:

```
from keras.models import load_model
from keras.preprocessing import image
```

In [73]:

```
model = load_model('conv.h5')
```

```
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM
Cloud account.
API_KEY = "BjvdOEwswTzK2Z89oJFJnYbSLF3G6NlwGBCK--ZzAZBw"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
```

```
# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values":
[array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/
045e3e38-87fa-4644-b8cc-36ffe866444a/predictions?version=2022-11-20',
json=payload_scoring,
 headers={'Authorization': 'Bearer ' + mltoken})
```

## 10.Advantages and Disdavantages :

### Advantages:

1.The communication between normal and deaf / dumb person becomes more effective.

2. There is no need of learning any sign language for communication.
3. The system designed is adaptive.
4. The communication system designed works efficiently on real time.
5. The designed system can be accessed easily on anywhere.

### Disadvantages:

1. The image should be captured correctly to get the accurate letter.
2. When the image is not captured fully, it might give the wrong letter as an output.
3. The speed of the system depends on the speed of the device in which it is used.
4. The accurcay of the system also depends on the network capacity of the device in which it is used.
5. The user must have a basic knowledge for effective use of the features of the application.

## 11. Conclusion:

This system mainly focus to establish effective communication between the deaf / dumb person. There is no need of learning any sign language for communication.

The use of the english language in effective manner is enough. This system is very useful in providing a proper conversation between a normal person and an impaired person.
The communication of impaired persons grows significantly. They can respond in a fast manner compared to the previous situations.

## 12. Future Scope:

Now, The Designed system identifies letters only. It's future enhancement is, it will convert the letters into words and form the sentence from these words and it is
converted into audio for blind people. Then by using the voice assistant it will describe the exact action to the impaired person.

## 13. Appendix:

### Source code:

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2,
zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

In [2]:

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
```

In [3]:

```
X_train = train_datagen.flow_from_directory('/content/Dataset/training_set',
target_size = (64,64), batch_size = 300 , class_mode = 'categorical',
color_mode = 'grayscale')
Found 15750 images belonging to 9 classes.
```

In [4]:

```
X_test = test_datagen.flow_from_directory('/content/Dataset/test_set',
target_size = (64,64), batch_size = 300, class_mode = 'categorical',
color_mode = 'grayscale')
Found 2250 images belonging to 9 classes.
```

In [5]:

```
model = Sequential()
model.add(Convolution2D(32,(3,3), input_shape = (64,64,1), activation =
'relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Flatten())
model.add(Dense(units = 512, activation = 'relu'))
```

```
model.add(Dense(units = 256, activation = 'relu'))
model.add(Dense(units = 128, activation = 'relu'))
model.add(Dense(units = 64, activation = 'relu'))
model.add(Dense(units = 9, activation = 'softmax'))
```

In [6]:

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

In [7]:

```
model.fit(X_train, steps_per_epoch = len(X_train),epochs=10, validation_data
= X_test, validation_steps= len(X_test))
Epoch 1/10
53/53 [==============================] - 20s 323ms/step - loss: 0.6905 -
accuracy: 0.7589 - val_loss: 0.2049 - val_accuracy: 0.9404
Epoch 2/10
53/53 [==============================] - 13s 246ms/step - loss: 0.0775 -
accuracy: 0.9773 - val_loss: 0.1749 - val_accuracy: 0.9738
Epoch 3/10
53/53 [==============================] - 13s 240ms/step - loss: 0.0266 -
accuracy: 0.9930 - val_loss: 0.2277 - val_accuracy: 0.9756
Epoch 4/10
53/53 [==============================] - 13s 241ms/step - loss: 0.0169 -
accuracy: 0.9954 - val_loss: 0.2426 - val_accuracy: 0.9782
Epoch 5/10
53/53 [==============================] - 13s 243ms/step - loss: 0.0113 -
accuracy: 0.9972 - val_loss: 0.2347 - val_accuracy: 0.9778
Epoch 6/10
53/53 [==============================] - 13s 244ms/step - loss: 0.0100 -
accuracy: 0.9973 - val_loss: 0.1532 - val_accuracy: 0.9809
Epoch 7/10
53/53 [==============================] - 13s 251ms/step - loss: 0.0065 -
accuracy: 0.9982 - val_loss: 0.2477 - val_accuracy: 0.9760
Epoch 8/10
53/53 [==============================] - 13s 239ms/step - loss: 0.0036 -
accuracy: 0.9989 - val_loss: 0.2555 - val_accuracy: 0.9782
Epoch 9/10
53/53 [==============================] - 13s 241ms/step - loss: 0.0037 -
accuracy: 0.9991 - val_loss: 0.2666 - val_accuracy: 0.9769
Epoch 10/10
53/53 [==============================] - 13s 244ms/step - loss: 0.0033 -
accuracy: 0.9991 - val_loss: 0.2840 - val_accuracy: 0.9769
```

Out[7]:

In [8]:

```
model.summary()
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        320

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )
```

```
 flatten (Flatten)           (None, 30752)             0

 dense (Dense)               (None, 512)               15745536

 dense_1 (Dense)             (None, 256)               131328

 dense_2 (Dense)             (None, 128)               32896

 dense_3 (Dense)             (None, 64)                8256

 dense_4 (Dense)             (None, 9)                 585

=================================================================
Total params: 15,918,921
Trainable params: 15,918,921
Non-trainable params: 0
_________________________________________________________________
```

In [9]:

```
model.save('Model.h5')
```

In [ ]:

**Github Repository Link:**

https://github.com/IBM-EPBL/IBM-Project-954-1658332450