PROJECT REPORT

REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

Industry Mentor(s):	Divya	**Faculty Mentor**:	Lakshmi Priya S
Team ID	PNT2022TMID53121	**Project ID**	16976-1659626137
Team Members:	Srivarsha Elangovan Vaseekaran S Srikanth S Iyer Vignesh G.R	**Roll Numbers:**	SSNCE195001109 SSNCE195001124 SSNCE195001108 SSNCE195001127

1.) INTRODUCTION

1.1) PROJECT OVERVIEW

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. A convolution neural network is used 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

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. 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.

2.) LITERATURE SURVEY

2.1) EXISTING PROBLEM

People constitute the backbone of a society and developing a system to regulate communication between any physically impaired individual and the regular person is a key area of study. The sign language that is been used by the deaf and dumb individuals can be converted into a form that is easily recognized by other individuals and vice versa thus helping in establishing smooth communication between them. These communication systems will play a major factor in determining the linguistic condition of country.

2.2) REFERENCES

[1] Ambavane, Pritesh & Karjavkar, Rahul & Pathare, Hemant & Relekar, Shubham & Alte, Bhavana & Sharma, Neeraj, “A Novel Communication System For Deaf And Dumb People using gesture”, ITM Web of Conferences, 2020, 32. 02003. 10.1051/itmconf/20203202003.
[2] Tinawi, Lillian & Harb, Reem & Nasser, Hassan-Roland & Zaylaa, Amira & Hamawy, Lara, “A New Dumb's Communication System”, 2017.
[3] Verma, Pallavi & Priyadarshani, Richa, “Design of Communication Interpreter for Deaf and Dumb Person”, International Journal of Science and Research, 2013, 4. 2640-2643.
[4] Aditi Kalsh and N.S. Garewa, “Sign Language Rcognition System,”,International Journal of Computational Engineering Research, Vol 03, Issue 6, June 2013.

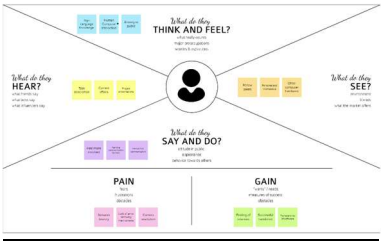
2.3) PROBLEM STATEMENT DEFENITION

Establishing transparent communication between the regular and the physically impaired people has been one of the key areas of research. By doing so, this will bring about a major change in the society. The conversion of sign language to speech and sound and vice versa encompasses different kinds of users to communicate with every other individual.

3.) IDEATION AND PROPOSED SOLUTION

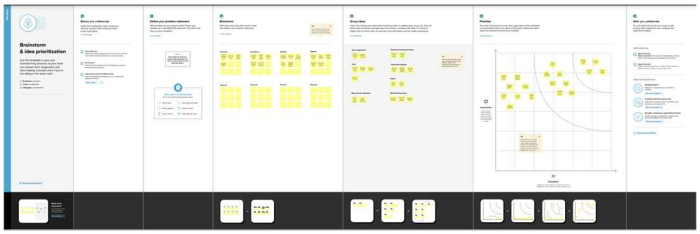
3.1) EMPATHY MAP CANVAS

Canvas Empathy Map.pdf **(Click on this link)**



3.2) IDEATION AND BRAINSTORMING

Ideation and Brainstorming.pdf **(Click on this link)**



3.3) PROPOSED SOLUTION

Proposed Solution.pdf **(Click on this link)**

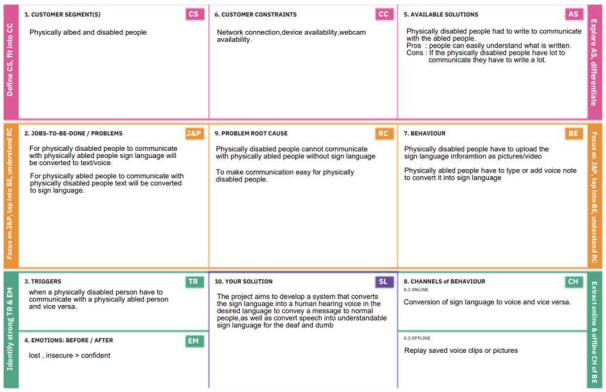
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	In our society, we have people with disabilities. The technology is developing day by day, but no significant developments are undertaken for the betterment of these people. Communications between deaf-dumb and a normal person has always been a challenging task. It is very difficult for deaf and dumb people to convey their message to normal people. Since normal people are not trained on hand sign language. In times of emergency, 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. Building a system with Voice Conversion combined 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.
2.	Idea / Solution description	The concepts of Machine learning algorithms and Neural networks were used to implement a solution. A Convolution Neural network is used to create a model that is subsequently trained on different hand gestures available in the dataset (almost around a thousand of them). All the hand gestures are fed into the model which are then processed, trained, and segregated using a certain machine learning algorithm. A certain No. of records are taken aside to continuously train the model and the rest are used to evaluate the learning of the built model. Clustering algorithms are used to segregate gestures into groups based on the different type of attributes available for a

		hand gesture. This information subsequently gets converted to human-understandable language and speech is given as output. In the other way, the same process is repeated where the input is given as a set of text commands from the end users, and they get converted to recognized hand gestures by the learning model which are then displayed to the impaired people. A Web application is built which uses this model. This Web application enables deaf and dumb people to convey their information
3.	Novelty / Uniqueness	A unique feature that sets apart this web application is that it not only provides a path for the impaired people to communicate comfortably, but also the regular people to communicate/respond back to them. The regular two-way form of communication is achieved in this process. The regular people make use of simple textual handwritten scripts to feed into the application as input, for which the impaired people receive the suitable hand gestures generated. In order to group all the available hand gestures based on attributes, Different clustering algorithm are used.
4.	Social Impact / Customer Satisfaction	The usage of this application creates a definite impact in the society. The end users consisting of the regular and impaired people are able to communicate comfortably without any form of hassle. Fear of Anxiety that always prevailed among them are substantially removed. There is no limitation in the process and the customers are satisfied with the system that is available.
5.	Business Model (Revenue Model)	Different strategies can be used to obtain financial benefit from the application. Initially, almost all of the comprising features can be made available free for the end users, for a specific time period. This will allow them to get acquainted with the software and the need to use it more and more. Selective features can be made as paid features as time progresses, by releasing 'Premium' or the more refined versions. Customer feedbacks can be collected on a regular basis since they constitute the heart

		of the application. They can be used to improve the system into more refined ones.
6.	Scalability of the Solution	There is a lot of potential for this application to expand and grow. The textual input that is received from the side of the regular users can be made to instead accommodate voice commands. This is an advancement of the previous feature, and this will allow surplus users to use the software. Though issues might be observed during the processing of these voice notes, implementation of the same would see a significant rise in the No. of users making use of the application.

3.4) PROBLEM SOLUTION FIT

Problem Solution Fit.pdf **(Click on this Link)**



4.) REQUIREMENT ANALYSIS

4.1) FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through Form Login through Gmail
FR-4	Image Upload	Image Upload via Drag & Drop Image Upload by choosing from local file subsystem
FR-5	Text Entry	Text Entry by Copy-Paste Text Entry by Typing
FR-6	Conversion	Sign-Language to Speech Conversion Text to Sign-Language Conversion
FR-7	Activity Log	View list of active users View log of translations performed

4.2) NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application can be used by clients without prior knowledge of machine-learning algorithms. Basic knowledge of text-entry and file directory subsystem is sufficient
NFR-2	**Security**	Access is provided only to registered users, either through a username-password pair or via their Gmail account
NFR-3	**Reliability**	The application has an extremely low failure rate / downtime and has a clear, predictable flow of operations
NFR-4	**Performance**	The application provides accurate translations from sign-language to speech as well as from text-to-sign language
NFR-5	**Availability**	Clients can use the application to perform conversions provided they have a stable Internet connection
NFR-6	**Scalability**	The application seamlessly and efficiently handles a growing number of clients / users without any significant reduction in performance

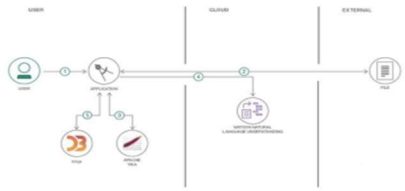
5.) PROJECT DESIGN

5.1) DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

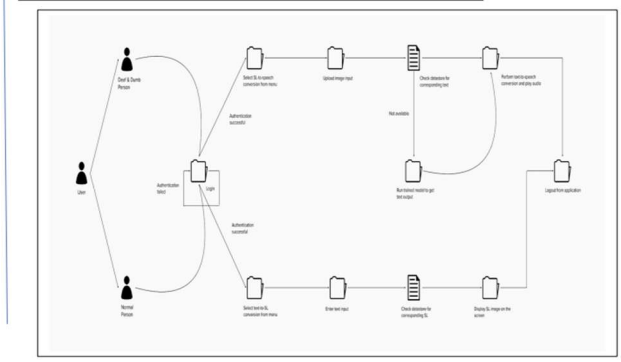
Example: (Simplified)

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)

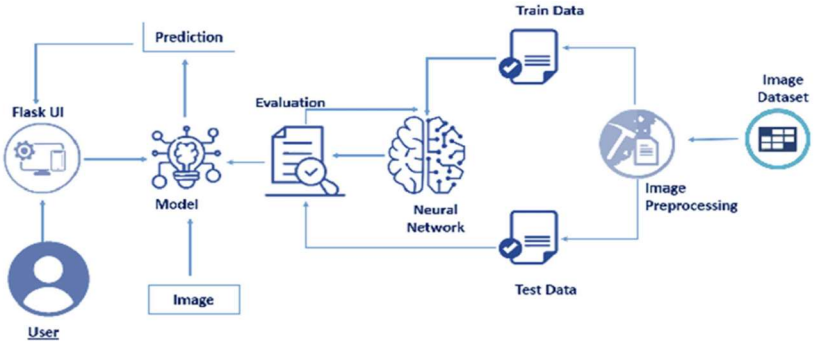


5.2) SOLUTION AND TECHNICAL ARCHITECTURE

Solution:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Technical Architecture:



5.3) USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User (Deaf & Dumb Person)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can access my account / dashboard via Gmail	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I will be redirected to my dashboard	High	Sprint-1
	Dashboard	USN-5	As a user, I can view a menu to perform sign-language to speech conversion	I will be redirected to the appropriate page on making my selection	High	Sprint-2
	Image Upload	USN-6	I will be provided with an option to upload an image of the sign-language in jpeg / png format	I will get a confirmation that the image has been uploaded successfully and can preview the image	High	Sprint-2
	Translation	USN-7	I will be provided with a button to initiate the translation process	I will be able to play an audio file containing the converted output	High	Sprint-3
User (Normal Person)	Registration	USN-8	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-9	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-10	As a user, I can register for the application through Gmail	I can access my account / dashboard via Gmail	Medium	Sprint-2
	Login	USN-11	As a user, I can log into the application by entering email & password	I will be redirected to my dashboard	High	Sprint-1
	Dashboard	USN-12	As a user, I can view a menu to perform sign-language to speech conversion	I will be redirected to the appropriate page on making my selection	High	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Text Entry	USN-13	I will be provided with a text box to type in the input	I can preview the entered text	High	Sprint-2
	Translation	USN-14	I will be provided with a button to initiate the translation process	I will be able to view an image containing the converted output	High	Sprint-3
Administrator	Login	USN-15	As an administrator, I can log into the application by entering email & password	I will be redirected to my dashboard	High	Sprint-1
	Dashboard	USN-16	As an administrator, I can view a log of the translations performed by all the users	I will be provided with an option to download a statement of the conversions performed	High	Sprint-3
		USN-17	As an administrator, I can view the list of users currently using the application	I can view a list of active users marked with a green dot preceding their name	High	Sprint-2

6.) PROJECT PLANNING AND SCHEDULING

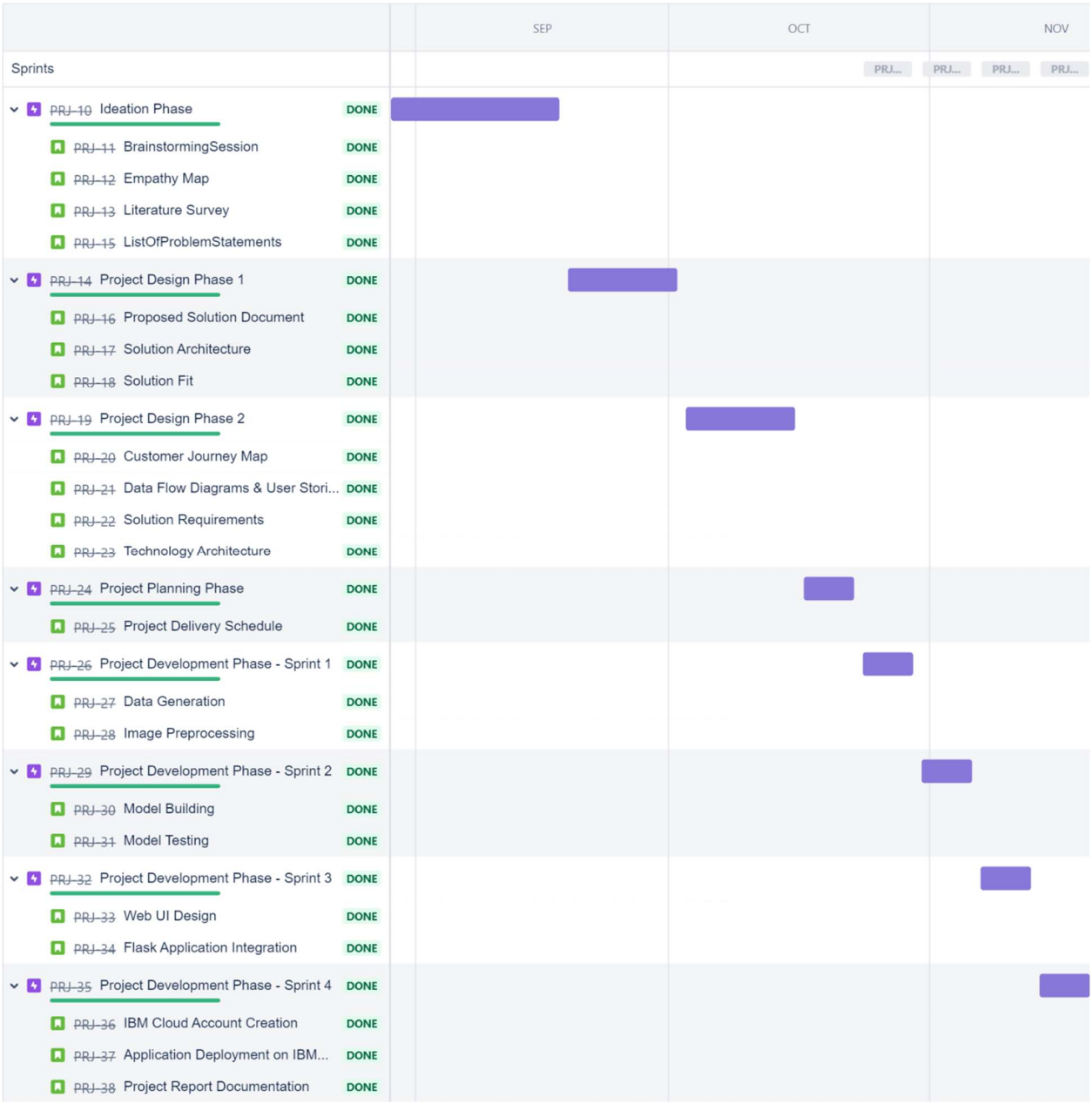
6.1) SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1 / USN-8	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Vaseekaran, Srivarsha
Sprint-1		USN-2 / USN-9	As a user, I will receive confirmation email once I have registered for the application	5	High	Srikanth, Vignesh
Sprint-2		USN-3 / USN-10	As a user, I can register for the application through Gmail	5	Medium	Srivarsha, Srikanth
Sprint-1	Login	USN-4 / USN-11 / USN-15	As a user, I can log into the application by entering email & password	10	High	Vaseekaran, Vignesh
Sprint-2	Dashboard	USN-5 / USN-12	As a user, I can view a menu to perform sign-language to speech conversion or vice versa	6	High	Vaseekaran, Srivarsha
Sprint-3		USN-16	As an administrator, I can view a log of the translations performed by all the users	10	High	Srikanth, Vignesh
Sprint-2		USN-17	As an administrator, I can view the list of users currently using the application	5	Medium	Srivarsha, Srikanth
Sprint-2	Image Upload	USN-6	I will be provided with an option to upload an image of the sign-language in jpeg / png format	5	High	Vaseekaran, Vignesh
Sprint-2	Text Entry	USN-13	I will be provided with a text box to type in the input	5	High	Vaseekaran, Vignesh
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-3	Translation	USN-7 / USN-14	I will be provided with a button to initiate the translation process	10	High	Srivarsha, Srikanth, Vaseekaran, Vignesh

6.2) SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	5 Nov 2022	20	5 Nov 2022
Sprint-3	20	6 Days	7 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	18 Nov 2022

6.3) REPORTS FROM JIRA



7.) CODING AND SOLUTIONING:

7.1) FEATURE 1 – Sign Language to Text Translation

The user can perform conversion from the sign of an ISL alphabet to the corresponding text. One has to simply upload image/photograph of the same and click a button, and the translated result is displayed on the screen.

7.2) FEATURE 2 – Text to Speech Conversion

The user also has the option to play the corresponding text aloud. This makes it easier for the other person to understand him/her easier and also models a real-life conversation much more closely.

8.) TESTING

8.1) TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
Conversion_TC_006	Functional	Conversion Page	Verify user is provided with an option to upload an image file		1. Click on button to perform conversion		Application provides an option to upload image file	Working as expected	Pass
Conversion_TC_007	Functional	Conversion Page	Verify user is provided with an option to perform conversion		1. Upload image file		Application provides a button to show corresponding text	Working as expected	Pass
Output_TC_008	Functional	Result Page	Verify converted text is displayed		1. Click on button to perform conversion		Application displays the converted text on the screen	Working as expected	Pass
Output_TC_009	Functional	Result Page	Verify output can be read aloud		1. Click on speaker icon on page		Application provides a clickable speaker icon	Working as expected	Pass
Output_TC_010	Functional	Result Page	Verify invalid input is handled		1. Upload an image file that is not a sign language in ISL		Application mentions that it is an invalid input	Working as expected	Pass

8.2) USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Real-Time Communication System Powered by AI for Specially Abled project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Flask Application	3	2	2	0	7
CNN Model	2	0	3	1	3
Web UI	5	5	0	0	10
Deployment	2	0	3	1	6
Subtotal	12	2	8	2	26

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
CNN Model	5	0	0	5
Flask Application/Web UI	10	0	0	10
Model deployed on IBM Watson Studio	10	0	0	10

9.) RESULTS

9.1) PERFORMANCE METRICS

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	A Convolution Neural Network model with 2 hidden layers that recognize the alphabet from a given ISL image input	PFB Figure 1
2.	Accuracy	Training Accuracy – 99.43% Validation Accuracy – 97.64%	PFB Figure 2

```python
cnn_model = Sequential() # Sequential Model

cnn_model.add(Conv2D(128, (3,3), activation='relu', input_shape=(128,128,1))) # Convolution Layer
cnn_model.add(MaxPooling2D(pool_size=(2,2))) # Pooling Layer

cnn_model.add(Conv2D(128, (3,3), activation='relu')) # Convolution Layer
cnn_model.add(MaxPooling2D(pool_size=(2,2))) # Pooling Layer

cnn_model.add(Flatten()) # Flatten Layer
cnn_model.add(Dense(128, activation='relu')) # Hidden Layer
cnn_model.add(Dense(128, activation='relu')) # Hidden Layer
cnn_model.add(Dense(9, activation='softmax')) # Hidden Layer
```

Python

Figure 1 : CNN Model

```
Epoch 1/3
158/158 [==============================] - 1964s 12s/step - loss: 0.2861 - accuracy: 0.9060 - val_loss: 0.2118 -
val_accuracy: 0.9644
Epoch 2/3
158/158 [==============================] - 1434s 9s/step - loss: 0.0269 - accuracy: 0.9912 - val_loss: 0.3211 -
val_accuracy: 0.9609
Epoch 3/3
158/158 [==============================] - 1474s 9s/step - loss: 0.0176 - accuracy: 0.9943 - val_loss: 0.3470 -
val_accuracy: 0.9764
```

Figure 2 : Model Accuracy

10.) ADVANTAGES AND DISADVANTAGES

Advantages:

a) The communication barrier between the physically impaired and regular people is broken down and is made a whole lot easier using the Sign language to Text/Voice conversion systems.

b) Building a model based on the different available hand signs and other features from the dataset provides more accuracy in the end results.

c) User friendly Web Interface for the end users thereby providing a customized experience.

d) The application is Scalable, i.e, more advancements and future works can be performed on the same underlying architecture.

Disadvantages

a) Prerequisite knowledge on the ISL (Indian Sign Language) is required to facilitate the communication between the users.
b) Cognitive emotions like Anger,Frustration,Anxiety,fear,etc will play an important role in the usability of the application.
c) The application may demand Internet and Webcam facilities, thereby requiring the users to have them as prerequisites.

11.) CONCLUSION

Communication between impaired people and normal people is exchanged in a smooth way. A system is developed 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. The system is built with the help of multiple Machine Learning Algorithms. The web interface for the application is built with the help of HTML, CSS, JS and Python Flask. PostGres database is used to store the details of users. A convolution neural network is built to create a model that is trained on different hand gestures. The system can be used by both normal and Impaired users to have hassle free communication between them. This is a significant advancement in the betterment of these people. The system was built after considerable amount of research in sign language and its working. The final system is deployed on IBM Watson and is available for use for all the users who want to make use of it for communication.

12) FUTURE SCOPE

There is a lot of potential for this application to expand and grow. The textual input that is received from the side of the regular users can be made to instead accommodate voice commands. This widens the scope of the project. This is an advancement of the previous feature, and this will allow surplus users to use the software. Though issues might be observed during the processing of these voice notes, implementation of the same would see a significant rise in the No. of users making use of the application. The dataset can also be constantly used and modified to add more sign expressions for further study of sign language analysis.

13) APPENDIX

Github Link:

https://github.com/IBM-EPBL/IBM-Project-16976-1659626137

Project Link:

https://dataplatform.cloud.ibm.com/ml-runtime/deployments/d45d9ae4-908b-4d66-9427-e0a6a07468e9?space_id=28bc8022-631a-417f-a9b4-9a242a292066&context=cpdaas

Source Code:

a. app.py

```
from flask import Flask,render_template,request,redirect
from flask_sqlalchemy import SQLAlchemy
import model
from gtts import gTTS
import os
from playsound import playsound
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] =
'postgresql://postgres:postgres@localhost:5432/realtimedb'
db = SQLAlchemy(app)
class Users(db.Model):
user_id = db.Column(db.Integer,primary_key = True,autoincrement=True)
email = db.Column(db.String(40),unique=True)
   password = db.Column(db.String(40))

   def __init__(self,email,password):
      self.email = email
      self.password = password
```

```
#db.create_all()



@app.route('/',methods=['POST','GET'])
def index():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        try:
            res = db.session.query(Users).filter(Users.email == email)
            for i in res:
                if(i.password == password):
                    return redirect('/dashboard')
                else:
                    return redirect('/')
        except:
            return redirect("/")
    return render_template('login.html')

@app.route("/signup",methods=['POST','GET'])
def signup():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        rePass = request.form['psw-repeat']
        if(password == rePass):
            user = Users(email,password)
            print(user)
```

```
            try:
                db.session.add(user)
                db.session.commit()
                return redirect('/')
            except:
                return redirect('/signup')
        else:
            return redirect('/signup')
    return render_template('signup.html')

@app.route("/dashboard")
def dashboard():
    return render_template("dashboard.html")
path = ''
res = ''
@app.route("/signtospeech",methods=['POST','GET'])
def signtospeech():
    global path
    global res
    if request.method == 'POST':
        path = request.form.get("image")
        print("path")
        print(path)
        res += str(model.predict(path))
        return render_template("output.html", img_path=str(path), pred=res)
    return render_template("signtospeech.html")

@app.route("/refresh",methods=['POST','GET'])
def refresh():
```

```
    global res
    res = ''
    if os.path.isfile('converted.mp3'):
        os.remove('converted.mp3')

@app.route("/audio",methods=['POST','GET'])
def audio():
    global res
    if os.path.isfile("converted.mp3"):
        playsound('converted.mp3')
    else:
        myobj = gTTS(text=res, lang='en', slow=False)
        myobj.save("converted.mp3")
        playsound('converted.mp3')

@app.route("/texttosign")
def texttosign():
    return "text to sign"

@app.route("/admindb")
def admindb():
    return "admin"

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

`

model.py

```
from keras_preprocessing import image
import numpy as np
from keras.models import load_model
import requests



cnn_model = load_model('trained_cnn_model.h5')



def predict_alpha(pos):
 res = {0:'A', 1:'B', 2:'C', 3:'D', 4:'E', 5:'F', 6:'G', 7:'H', 8:'I'}
 return res[pos]

def predict(img_src):
 img = image.load_img(img_src, target_size=(128,128), color_mode='grayscale')
 arr = image.img_to_array(img)
 arr = np.expand_dims(arr, axis=0)
 API_KEY = "uK6_2BkDeaVcCaMnVSMABbOrpSde4G6uMNgSkXkA55e9"
 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}
 payload_scoring = {"input_data": [{"values": arr.tolist()}]}
```

```
 response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/d45d9ae4-908b-4d66-9427-
e0a6a07468e9/predictions?version=2022-11-16', json=payload_scoring,
 headers=header)
 print(type(response_scoring))
 print(response_scoring.json())
 print("Scoring response")
 res  = response_scoring.json()
 idx = res["predictions"][0]["values"][0][1]
 #prediction = np.argmax(cnn_model.predict(arr))
 return predict_alpha(idx)
```