Real-Time Communication System Powered by AI for Specially Abled:

Abstract:

In this paper, we discuss a stand-alone technology that would make it simple and fluid for hearing-impaired and normal individuals to converse with one another. We provide an application for automatically translating visual data into text in real time while using image processing to recognise American Sign Language. Video footage from a digital camera or camera application will be used to create a real-time hand gesture detection system after which the hand position and location will be tagged and isolated via cropping. The hand motions will then be identified by image processing and compared to a gesture database that has already been created, which will be utilised for text conversion on the screen. Additionally, the programme allows regular users to write the text down and exhibit the corresponding animation of hand motions. This system does textual representation and real-time recognition of American Sign Language, producing more accurate results in the shortest amount of time. It won't just help the specially abled; it may also be applied in a number of different technological contexts.

Additionally, this method gives users the freedom to study American Sign Language at their own speed, whenever they want, anywhere—at home or at work.

Real-Time Communication System Powered by AI for Specially Abled

Submitted by :

TEAM ID:- PNT2022TMID36394

MITHOON.N.S (110819104301)

SHYAM.M.V (110819104303)

SIDDHARTH.V (110819104033)

TANMOY DAS (110819104304)

1. INTRODUCTION:

Project Overview:

By discussing their ideas, opinions, and experiences with others around them, people come to know one another. There are several methods to do this, but the gift of "Speech" is the finest. Speech allows everyone to communicate their ideas and comprehend one another quite well. It would be unfair to ignore those who are denied this wonderful gift: individuals with disabilities. In these circumstances, sign language has traditionally been used using the human hand.

The most common issue for those with hearing/speech impairments is being unable to communicate with others. They utilise Sign Language to communicate with others in order to express their thoughts or feelings (SL). Sign language (SL) is a prominent way of communication mechanism utilised regularly by persons who are deaf or hard of hearing. This nonverbal language employs hand motions as well as occasional face gestures.

With the advancement of technology, some form of device or instrument that can mediate between hard-of-hearing people and normal people is necessary, so that they may easily interact with each other without the need for a third person as an interpreter.

Purpose :

In order to communicate with regular people, the project intends to create a system that translates sign language into text that is legible by humans. A convolution neural network is being used to build a model that is trained on various hand motions. A web application utilising this concept is created. With the use of this software, persons who are deaf or dumb may communicate using signs that are translated into language that is intelligible to others

2. Literature Survey

Title-1: Artificial intelligence is like artificial god for specially abled
Author: Prem Mohan
Project Description: Artificial intelligence is not designed to replace humans but rather to enhance our lives by helping us do things we are unable to do on our own. Many companies are working on this type of research, including Google Deepmind, IBM Watson, Apple Siri, Microsoft Cortana, etc., which means there will likely be many new developments soon. These innovations could positively impact everyone's life – even those without disabilities – because they make everyday tasks easier and less time-consuming.
Drawbacks: AI for Accessibility program uses the potential of Artificial Intelligence to develop solutions to many physical and cognitive challenges disabled individuals face at work and in daily life to promote social

inclusion for them

Title -2: Communication skills in the disabled
Authors: Liubov Ben-Noun
Project Description: Communication is an important human characteristic. In order to maintain relationships effectively humans must communicate with each other. In everyday life, there are a variety of communications including with work colleagues, family, neighbours, and friends, some efficient and some inefficient. Drawbacks: Ethics and morality are important human features that can be difficult to incorporate into an AI. The rapid progress of AI has raised a number of concerns that one day, AI will grow uncontrollably, and eventually wipe out humanity

Title -3: Sign Language Recognition System for People with Disability using Machine Learning and Image Processing
Authors: Bayan Mohammed Saleh, Muhammad Usman Tariq
Project Description: Communication creates bonding and relations among the people, whether persons, social, or political views.Most people communicate efficiently without any issues, but many cannot due to disability. They cannot hear or speak, which makes Earth a problematic place to live for them. Even simple basic tasks become difficult for them. Disability is an emotive human condition. It limits the individual to a certain level of performance. Being deaf and dumb pushes the subject to oblivion, highly introverted. The application of technology should create a platform or a world of equality despite the natural state of humans. Communication, D-talk is a system that allows people who are unable to talk and hear be fully understood and for them to learn their language easier and also for the people that would interact and communicate with them.
Drawbacks: Sign language allows deaf and hard of hearing people to communicate quickly and effectively with others who use sign language.

Title -4: Real-Time Communications in the Web: Issues, Achievements, and Ongoing Standardization Efforts
Authors: Salvatore Loreto, Simon Pietro Romano
Project Description: Web Real-Time Communication (WebRTC) is an upcoming standard that aims to enable real-time communication among Web browsers in a peer-to-peer fashion. The IETF RTCWeb and W3C WebRTC working groups are jointly defining both the APIs and the underlying communication protocols for setting up and managing a reliable communication channel between any pair of next-generation Web browsers. Drawbacks: Asynchronous communication takes place when there is a delay or lag between sharing information and receiving a response

Problem Statement:

To develop an automatic sign language recognition system with the help of image processing and computer vision techniques. To use natural image sequences, without the signer having to wear data gloves or colored gloves, and to be able to recognize hundreds of signs. The motivation for this work is to provide a real time interface so that signers can easily and quickly communicate with non-signers. To efficiently and accurately recognize signed words, using a minimal number of training examples.

3. Requirements

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 Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	User Selection	User can select option either sign language to text/voice or text/voice to sign language
FR-4	User Requirement	Mobile or Laptop
FR-5	User Guide	Guidance to use the application will be inbuilt

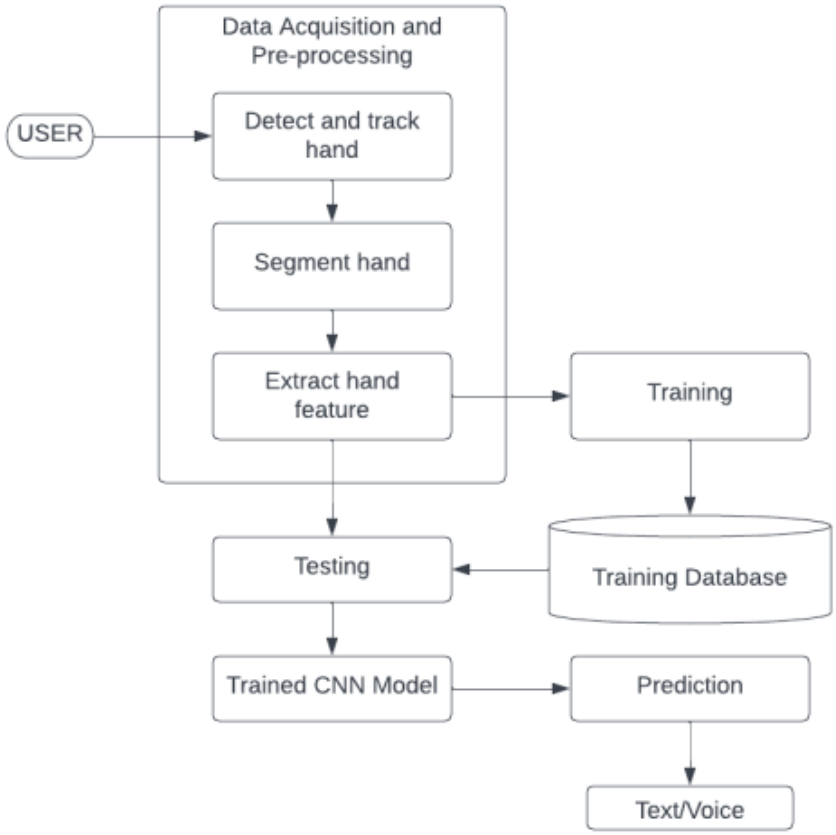
Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This application helps to convert the sign language to text/voice or vice versa. So this application will be useful for both specially abled people and normal people to communicate each other.
NFR-2	**Security**	This system is protected and only authorized users can access it.
NFR-3	**Reliability**	The application will predict the hand gesture and convert it to exact text or voice.
NFR-4	**Performance**	The application responds to a user in seconds and the hardware and software works well.
NFR-5	**Availability**	It is accessible by authorised user from anywhere at any time whenever they need.
NFR-6	**Scalability**	It can predict different types of sign language at a time. More numbers of users can be accessed.

4. Project Design

Data Flow Diagrams:



Block Diagram:

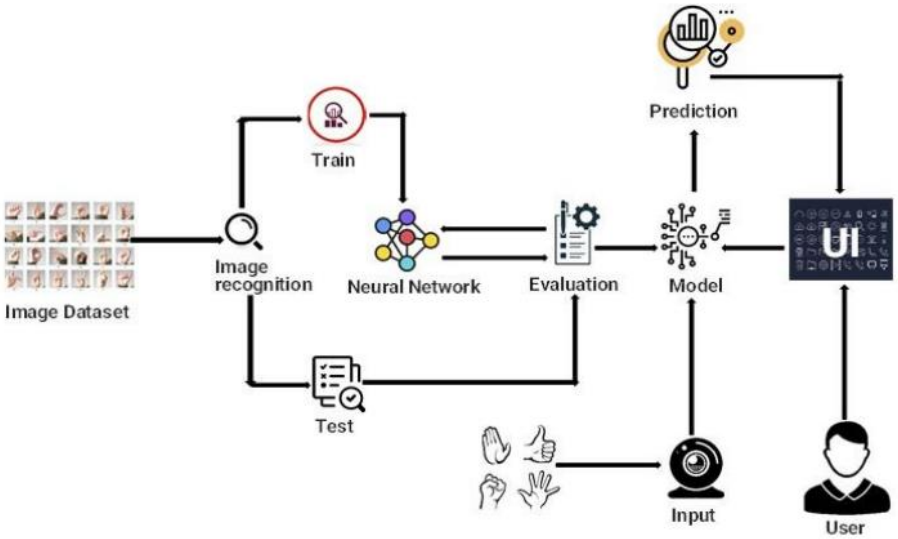


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	By using Web UI and Mobile app user interacts with the application	HTML, CSS, JavaScript etc.
2.	Application Logic-1	Getting the Hand Gesture Images dataset	Python
3.	Application Logic-2	Analysing the application and the representation of the hand gestures	IBM Watson, CNN
4.	Application Logic-3	Getting audio as input data	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	NoSQL
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant
7.	File Storage	Received hand gesture movements and the audio speech is stored in the cloud	IBM Block Storage or Other Storage Service or Local Filesystem
8.	CNN	Purpose of CNN is used for the understanding of sign to human readable language and vice-versa	CNN, Object detection model, NLP
9.	Machine Learning Model	AI-Machine Learning model is used for the identification of hand gestures recognition, Sign language and vice-versa	Object Recognition Model, CNN and NLP for voice data and hand gestures
10.	Infrastructure (Server / Cloud)	Deploying the AI and CNN model using flask in the web application of cloud server	Python Flask

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks which are used	TensorFlow, RNN, Pytorch
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Firewall and some security related to software.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	3-tier architecture
4.	Availability	Availability of application	Image recognition, gesture recognition, text and voice recognition.
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache)	Using CNN, Machine Learning for conversion which the performance will be good

- # User Stories

USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user, I can get all service and access in dashboard.	I can get all service access in dashboard	Medium	Sprint-3
Customer (Web user)	Registration	USN-7	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-8	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
	Login	USN-9	As a user, I can log into the application by entering email & password		High	Sprint-1
Customer Care Executive	Service	USN-10	As a user, I can avail the service by calling customer care or through mail.	I can avail the service by calling customer care or through mail.	Medium	Sprint-2
Administrator	Manage	USN-11	As a Admin, I can manage the database server and application tools.	I can manage the database server and application tools.	High	Sprint-2

5. Coding, Solutioning and Testing

Image processing:

```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```python
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
```

```python
test_datagen = ImageDataGenerator(rescale=1/255)
```

```python
import tensorflow as tf
import os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import IPython.display as display
from PIL import Image
import pathlib
```

```python
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```python
train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True, vertical_flip=False)
```

```python
test_datagen= ImageDataGenerator(rescale=1./255)
```

```python
x_train = train_datagen.flow_from_directory('/content/drive/MyDrive/IBM_nalaiyathiran/Dataset/training_set',target_size=(64,64), batch_size=300,
                                            class_mode='categorical', color_mode = "grayscale")
```

```
Found 15750 images belonging to 9 classes.
```

```python
x_test = test_datagen.flow_from_directory('/content/drive/MyDrive/IBM_nalaiyathiran/Dataset/test_set',target_size=(64,64), batch_size=300,
                                          class_mode='categorical', color_mode = "grayscale")
```

```
Found 2250 images belonging to 9 classes.
```

```python
x_train.class_indices
```

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

```python
x_test.class_indices
```

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

Model Building:

Loading the Dataset & Image Data Generation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Python

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

Python

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'C:\Users\NAGARAJAN K\Desktop\Mithoon IBM project\Development Phase\Data Collection\training_set',target_size=(64,64), class_mode='categorical',batch_s
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'C:\Users\NAGARAJAN K\Desktop\Mithoon IBM project\Development Phase\Data Collection\test_set',target_size=(64,64), class_mode='categorical',batch_size=90
```

Python

```
Found 15750 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

Python

```
Len x-train :  18
Len x-test :  3
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

Python

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

Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

Python

```
# Creating Model
model=Sequential()
```

Python

```
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

Python

```
# Adding Pooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
```

Python

```python
# Adding Flattem layer
model.add(Flatten())
```

Python

```python
# Adding Dense Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
```

Python

```python
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Python

```python
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

Python

```
C:\Users\NAGARAJAN K\AppData\Local\Temp\ipykernel_21636\1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))

Epoch 1/10
18/18 [==============================] - 264s 15s/step - loss: 1.2747 - accuracy: 0.5844 - val_loss: 0.4318 - val_accuracy: 0.8951
Epoch 2/10
18/18 [==============================] - 78s 4s/step - loss: 0.2923 - accuracy: 0.9161 - val_loss: 0.2998 - val_accuracy: 0.9267
Epoch 3/10
18/18 [==============================] - 78s 4s/step - loss: 0.1279 - accuracy: 0.9630 - val_loss: 0.2047 - val_accuracy: 0.9547
Epoch 4/10
18/18 [==============================] - 78s 4s/step - loss: 0.0635 - accuracy: 0.9834 - val_loss: 0.1839 - val_accuracy: 0.9613
Epoch 5/10
18/18 [==============================] - 79s 4s/step - loss: 0.0407 - accuracy: 0.9894 - val_loss: 0.2094 - val_accuracy: 0.9684
Epoch 6/10
18/18 [==============================] - 78s 4s/step - loss: 0.0261 - accuracy: 0.9935 - val_loss: 0.1422 - val_accuracy: 0.9724
Epoch 7/10
18/18 [==============================] - 77s 4s/step - loss: 0.0166 - accuracy: 0.9968 - val_loss: 0.1667 - val_accuracy: 0.9720
Epoch 8/10
18/18 [==============================] - 79s 4s/step - loss: 0.0131 - accuracy: 0.9973 - val_loss: 0.1798 - val_accuracy: 0.9751
Epoch 9/10
18/18 [==============================] - 77s 4s/step - loss: 0.0107 - accuracy: 0.9981 - val_loss: 0.1541 - val_accuracy: 0.9791
Epoch 10/10
18/18 [==============================] - 80s 4s/step - loss: 0.0077 - accuracy: 0.9987 - val_loss: 0.1743 - val_accuracy: 0.9778

<keras.callbacks.History at 0x2d9d47ab8b0>
```

Saving the Model

```python
model.save('asl_model-36394.h5')
```

Testing the model

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

Python

```
model=load_model(r'C:\Users\NAGARAJAN K\Desktop\Mithoon IBM project\Development Phase\Sprint 2\asl_model-36394.h5')
img=image.load_img(r'C:\Users\NAGARAJAN K\Desktop\Mithoon IBM project\Development Phase\Data Collection\test_set\A\17.png',
                   target_size=(64,64))
```

Python

```
img
```

Python

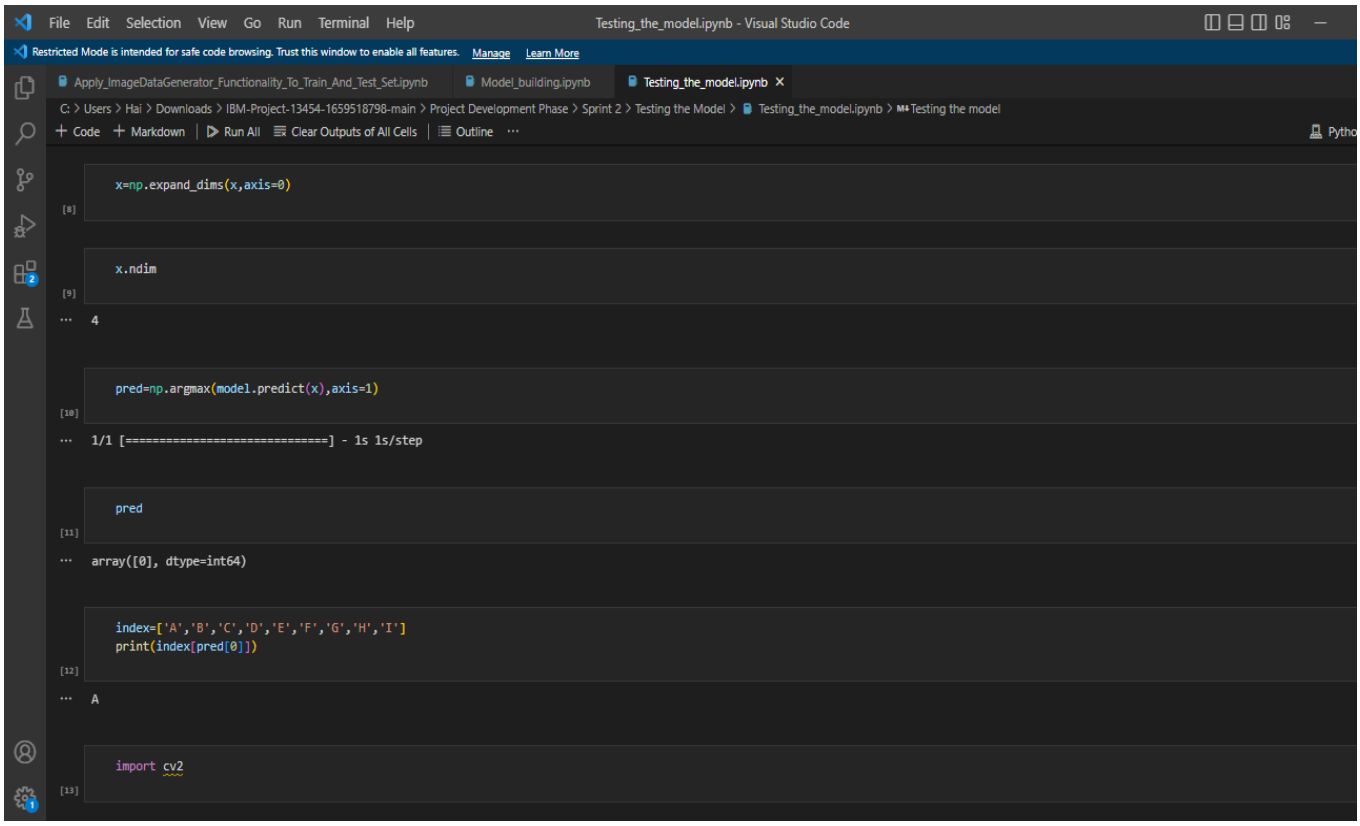
```
x=image.img_to_array(img)
```

Python

```
x.ndim
```

Python

```
3
```



```python
img=cv2.imread(r'C:\Users\NAGARAJAN K\Desktop\Mithoon IBM project\Development Phase\Data Collection\test_set\A\17.png',1)
```

```python
img1=cv2.imread(r'C:\Users\NAGARAJAN K\Desktop\Mithoon IBM project\Development Phase\Data Collection\test_set\A\17.png',0)
```

```python
print(img.shape)
```

```
(64, 64, 3)
```

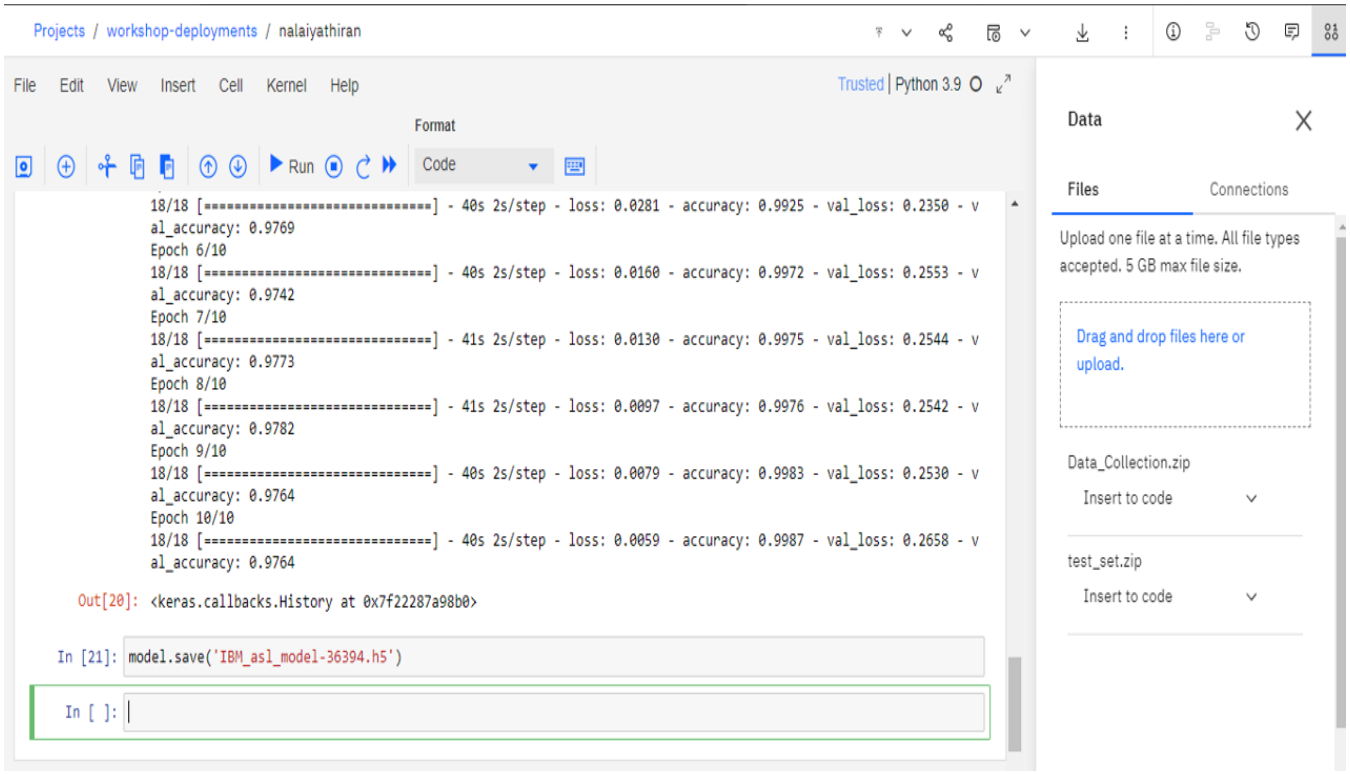
```python
import cv2
cv2.imshow('',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```python
```

Training Model:

```python
from function import *
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.callbacks import TensorBoard
label_map = {label:num for num, label in enumerate(actions)}
# print(label_map)
sequences, labels = [], []
for action in actions:
    for sequence in range(no_sequences):
        window = []
        for frame_num in range(sequence_length):
            res = np.load(os.path.join(DATA_PATH, action, str(sequence), "{}.npy".format(frame_num)))
            window.append(res)
        sequences.append(window)
        labels.append(label_map[action])

X = np.array(sequences)
y = to_categorical(labels).astype(int)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05)

log_dir = os.path.join('Logs')
tb_callback = TensorBoard(log_dir=log_dir)
model = Sequential()
model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(30,63)))
model.add(LSTM(128, return_sequences=True, activation='relu'))
model.add(LSTM(64, return_sequences=False, activation='relu'))
model.add(Dense(64, activation='relu'))#MithoonNS
model.add(Dense(32, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))
res = [.7, 0.2, 0.1]

model.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['categorical_accuracy'])
model.fit(X_train, y_train, epochs=200, callbacks=[tb_callback])
model.summary()

model_json = model.to_json()
with open("model.json", "w") as json_file:
    json_file.write(model_json)
model.save('asl_model-36394.h5')
```

Train Model On IBM:



Spyder Deployment Code:

app.py 9+ × main.py 9+

C: > Users > NAGARAJAN K > Desktop > Mithoon_IBM_NalaiyaThiran > app.py > generate_frames

```python
from flask import Flask,render_template,Response
import cv2
import mediapipe as mp
from function import *
from keras.utils import to_categorical
from keras.models import model_from_json
from keras.layers import LSTM, Dense
from keras.callbacks import TensorBoard
import numpy as np
#import actions

app=Flask(__name__)
camera=cv2.VideoCapture(0)

def generate_frames():
    while True:

        ## read the camera frame
        success,frame=camera.read()
        if not success:
            break
        else:
            json_file = open("model.json", "r")
            model_json = json_file.read()
            json_file.close()
            model = model_from_json(model_json)
            model.load_weights("asl_model-36394.h5")

            colors = []
            for i in range(0,20):
                colors.append((255,255,255))
            print(len(colors))
            def prob_viz(res, actions, input_frame, colors,threshold):
                output_frame = input_frame.copy()
                for num, prob in enumerate(res):
                    cv2.rectangle(output_frame, (0,60+num*40), (int(prob*100), 90+num*40), colors[num], -1)
                    cv2.putText(output_frame, actions[num], (0, 85+num*40), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255), 2, cv2.LINE_AA)
```

```
38
39                  return output_frame
40
41              sequence = []
42              sentence = []
43              accuracy=[]
44              predictions = []
45              threshold = 0.8
46
47
48              with mp_hands.Hands(
49                  model_complexity=0,
50                  min_detection_confidence=0.5,
51                  min_tracking_confidence=0.5) as hands:
52
53
54                      ret, frame = camera.read()
55                      cropframe=frame[40:400,0:300]
56                      frame=cv2.rectangle(frame,(0,40),(300,400),255,2)
57                      image, results = mediapipe_detection(cropframe, hands)
58
59                      keypoints = extract_keypoints(results)
60                      sequence.append(keypoints)
61                      sequence = sequence[-30:]
62
63                      try:
64                          if len(sequence) == 30:
65                              res = model.predict(np.expand_dims(sequence, axis=0))[0]
66                              print(actions[np.argmax(res)])
67                              predictions.append(np.argmax(res))
68
69
70                              if np.unique(predictions[-10:])[0]==np.argmax(res):
71                                  if res[np.argmax(res)] > threshold:
72                                      if len(sentence) > 0:
73                                          if actions[np.argmax(res)] != sentence[-1]:
74                                              sentence.append(actions[np.argmax(res)])
```

```
                except Exception as e:
                    pass

                cv2.rectangle(frame, (0,0), (0, 0), (24, 117, 16), -1)
                cv2.putText(frame,"Predicted Sign:     "+' '.join(sentence)+"-"+"-".join(accuracy), (3,30),
                            cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 2, cv2.LINE_AA)

            ret,buffer=cv2.imencode('.jpg',frame)
            frame=buffer.tobytes()

        yield(b'--frame\r\n'
                b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')


@app.route('/')
def index():
    return render_template('index.html')

@app.route('/video')
def video():
    return Response(generate_frames(),mimetype='multipart/x-mixed-replace; boundary=frame')

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

    if request.method == 'POST':
        the_question = request.form['question']

        response = processor.chatbot_response(the_question)

    return jsonify({"response": response })


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

6. Result

Snapshots of our model predicting the hand gestures:


SIGN LANGUAGE DETECTION
ABOUT US
Predicted Sign: C-99.92274045944


SIGN LANGUAGE DETECTION
ABOUT US
Predicted Sign: A-93.40398311614

7. Advantages and Disadvantages :

Advantages:

1. It is feasible to develop a mobile application to close the communication gap between the hearing-impaired and the rest of society.
2. The user may select which sign language to read by adding the dataset when new sign language standards are created.
3. The disabled people who have hearing impairment will not have to be socially anxious anymore. They can communicate with great confidence

Disadvantages:

1. The present model is limited to the letters A through I.
2. Alphabets from J cannot be recognised in the absence of gesture recognition because they need user input in the form of a gesture.
3. The accuracy isn't excellent because there aren't many or high-quality photographs in the dataset, but that can be fixed by changing the datas.

8. Conclusion :

The use of sign languages can help normal and deaf individuals communicate more effectively. Our approach strives to reduce the communication gap between the deaf community and the rest of society since it supports two-way conversation. Our technology converts sign languages into human-understandable English language. With the help of this technology, the model receives hand gestures, recognises them, and then shows the corresponding Alphabet on the screen. This initiative allows deaf-mute persons to perform sign language with their hands, which will later be translated into alphabets.

9. Future Scope :

For persons with particular needs, such as the deaf and dumb, having technology that can convert hand sign language to its appropriate alphabet is a game changer. The web programme may easily be developed to detect letters other than "I," numbers, and other symbols with the addition of gesture recognition. Gesture recognition can also be used to control software and hardware interfaces.

10. References

[1] Keras Image Processing Doc :- https://keras.io/api/preprocessing/image/

[2] Keras ImageDataset From Directory Doc :- https://keras.io/api/preprocessing/image/#imagedatasetfromdirectory-function

[3] CNN using Tensorflow :- https://www.youtube.com/watch?v=umGJ30-15_A

[4] OpenCV Basics of Processing Image :- https://www.youtube.com/watch?v=mjKd1Tzl70I

[5] Flask Basics :- https://www.youtube.com/watch?v=lj4I_CvBnt0

[6] IBM Academic Partner Account Creation :- https://www.youtube.com/watch?v=x6i43M7BAqE

[7] CNN Deployment and Download through IBM Cloud :- https://www.youtube.com/watch?v=BzouqMGJ41k

[8] Matusiak, K., Skulimowski, P., & Strumiłło, P. (2013, June). Object recognition in a mobile phone
application for visually impaired users. In 2013 6th International Conference on Human System Interactions (HSI) (pp. 479-484). IEEE
.
[9] Hermus, K., & Wambacq, P. (2006). A review of signal subspace speech enhancement and its application
to noise robust speech recognition. EURASIP Journal on Advances in Signal Processing, 2007(1), 045821.

[10] Dimitrov, V., Jullien, G., & Muscedere, R. (2017). Multiple-base number system: theory and applications.
CRC press.
.
[11] Huyan, Z., Xu, L., Fang, S., Liu, Z., Zhang, X., & Li, L. (2014). Field information acquisition system
research based on offline speech recognition. Int. J. Database Theory Appl, 7, 45-58.

[12] Bigham, J. P., Jayant, C., Miller, A., White, B., & Yeh, T. (2010, June). VizWiz:: LocateIt-enabling blind people to locate objects in their environment. In 2010 IEEE Computer Society Conference on Computer
Vision and Pattern Recognition-Workshops (pp. 65-72). IEEE.

[13] Manduchi, R., Kurniawan, S., & Bagherinia, H. (2010, October). Blind guidance using mobile computer
vision: A usability study. In Proceedings of the 12th international ACM SIGACCESS conference

on
Computers and accessibility (pp. 241-242).

[14] Ivanchenko, V., Coughlan, J., Gerrey, W., & Shen, H. (2008, October). Computer vision-based clear path
guidance for blind wheelchair users. In Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility (pp. 291-292).

[15] Johnsen, A., Grønli, T. M., & Bygstad, B. (2012). Making touch-based mobile phones accessible for the
visually impaired. Norsk informatikkonferanse,(Bodø, Norway, 2012).

[16] Jiang, R., Lin, Q., & Qu, S. (2016). Let Blind People See: Real-Time Visual Recognition with Results
Converted to 3D Audio. Report No. 218, Standord University, Stanford, USA.

[17] Kamble, K., & Kagalkar, R. (2014). A review: translation of text to speech conversion for Hindi
language. International Journal of Science and Research (IJSR) Volume, 3.

[18] Kumar, A., & Chourasia, A. (2018). Blind Navigation System Using Artificial Intelligence. International
Research Journal of Engineering and Technology, 5(3).

[19] BELGHIT, H., & BELLARBI, A. Object Recognition Based on ORB Descriptor for Markerless
Augmented Reality.

[20] Coughlan, J., & Manduchi, R. (2009). Functional assessment of a camera phone-based wayfinding system operated by blind and visually impaired users. International Journal on Artificial Intelligence
Tools, 18(03), 379-397.

[21] Chen, C., & Raman, T. V. (2009). Announcing eyes-free shell for Android. Retrieved December, 21, 2016.

[22] Gill, J. (2000). Personal electronic mobility devices. Information for Professionals Working with Visually
Disabled People. http://www. tiresias. org.

[23] Coughlan, J., & Manduchi, R. (2007). Color targets: Fiducials to help visually impaired people find their
way by camera phone. EURASIP Journal on Image and Video Processing, 2007, 1-13.

[24] Arora, S. J., & Singh, R. P. (2012). Automatic speech recognition: a review. International Journal of
Computer Applications, 60(9).

[25] Omankhanlen, A. E., & Ogaga-Oghene, J. (2013). The Dynamics of Global Strategy and Strategic
Alliances in International Trade and Investment. INTERNATIONAL JOURNAL OF RESEARCH IN
COMPUTER APPLICATION & MANAGEMENT, 3(12), 41-48.

- **Github link :**
  https://github.com/IBM-EPBL/IBM-Project-13454-1659518798