REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLYABLED

CHAPTER 1

INTRODUCTION

a. 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 trainedon hand sign language. In emergency times conveying theirmessage is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired personin any language.

The project aims to develop a system that converts the sign language into a humanhearing 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 image is given asoutput.

b. PURPOSE

People get to know one another by sharing their ideas, thoughts, and experienceswith those around them. There are numerousways to accomplish this, one of which is the gift of images. Everyone can very convincingly transfer their thoughts and understand each other through images. It will be unjust if we overlook those who are denied this priceless gift: the deaf and dumb. In such cases, the human hand has remained the preferred method of communication. The project's purpose is to create a system that translates sign language into a human understandable language so that ordinary people may understand it.

CHAPTER 2 LITERATURE SURVEY

a. EXISTING PROBLEM

Some of the existingsolutions for solving this problem are:

1. Face Based Real Time Communication for Physically and Speech Disabled People

An improved real-time communication system using machine learning and computer vision. The aim is to create a communication channel between the specially a bled and the society, so they can express there feelings, thoughts and understand otherpeople's feelings and thoughts throughreal time communication and facial expressions.

2. Artificial Intelligence and Accessibility

Seeing AI, visually impaired people can easily read their mail by

placing documents under the smartphone camera. AI technology can apply to any type of disability profile. For instance,people with reducedmobility can control everything athome.

3. Survey on application of Artificial Intelligence in Cyber Security

Cyber security refers to protecting your personal computerfrom malicious software. Machine learning has a lot many algorithms and system which protect users from threats. Such as the Pay pal app which was developed in December 1998, uses machinelearning algorithms to protect its users from different threatsand online spoofing.It uses three types of machine learningalgorithms that are linear, neuralnetwork and deep learningalgorithm.

4. Machine Learning based techniques in data analysis

It is an application from which we can virtually explore streets of cities. It uses a dense geo samplingtool to shows the streetsof cities. Streetsare captured througha fleet of vehicles equipped with a specialized camera. After collection of photos, they are digitally processedand combined togetherand looks like a single image. From files reported for privacy, Google pixelated faces of pedestrian and license plate which iscaptured. Web mappingtechnologies have been embraced by discipline such as

geography, archaeology and ecology, but also by several social scientific disciplines. Researchers working in the discipline of geography, archaeology, and ecology quickly incorporated web based mapping technologies into their research designs. There are various applications

of google street view in research field, although the number stillremains limited. It is also used for better estimation of fish catching, estimation of forestrybiomass in India, estimation of area of different regionsor lakes, etc.

b. References

i. Ann, O. C., Lu, M. V., & Thing, L. B. (2011). A face based real time communicationfor physically and speech disabled people. In Assistive and Augmentative Communication for the Disabled: Intelligent Technologies for Communication, Learningand Teaching (pp. 70-102). IGI Global.

ii. Azmi, A., Alsabhan, N. M., & AlDosari, M. S. (2009). The Wiimote with SAPI: Creating an accessible low-cost, human computer interface for the physically disabled. International Journal of ComputerScience and NetworkSecurity, 9(12), 63-68.

iii. Li, J. H. (2018).Cyber security meets artificial intelligence: a survey.Frontiers ofInformation Technology & Electronic Engineering, 19(12), 1462-1474.

iv. Machine Learning based techniques in data analysis (Lavanya Vemulapalli,Dr.P.Chandra Sekhar – 2018)

C. Problem Statement Definition

Theproject aims to develop a system that converts the sign languageinto a human hearing voice in the desired language to convey a message to normal people, as well as convertspeech 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 differenthand

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 images are given as output.

Example:

Problem Statement (PS)	**I am (Customer)**	**I'm tryingto**	**But**	**Because**	**Which makes me feel**

PS-1	Person withoutdisabi lity	Communica te with deaf and dumb viaphone	It is notposs ib le	I can't ableto understa nd the signlangua ge	Frustrat ed
PS-2	A human like everyone	Communica tefreely with othe rs	I cannot do so	I am a deaf/du mb	Cap vated as well as unmo vated

CHAPTER 3 IDEATION& PROPOSED SOLUTION

a.

Empathy Map Canvas

b. **Ideation & Brainstorming**

C. Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to besolved)	Sign Language is a communication method used bypeople with hearingdisability or speaking disabilityusing hand gesture. Since it is not know by everyone people with disabilitytents to face difficulty while communicating. The proposed system is to help them convert the gesture-based sign language to voice based message
2.	Idea / Solution description	The idea is to identify the live gestured basedsign language and to translate it into the voice- based message to make the communication easy for thedifferently abled people
3.	Novelty / Uniqueness	The idea is to createa system thatwill ease outthe processes of conversion of sign language tohearable voice message. The application is trained withevery gesturepossible.
4.	Social Impact/Customer Satisfaction	1. To boost the confidence of a differently abledperson by making them independent 2. To breakthe communication barrier 3. Ease the conversion of sign language tovoice-based message 4. To improvethe everyday livesofpeoplewithdisabilities
5.	Business Model(Revenue Model)	The targeted customers of this system are people with hearing disability and speaking disability and the people around them Because of its uniquenessand the essentiality undoubtedly the market of thesystem will be huge

6.	Scalability of the Solution	The proposed application for the people with disability is accessible in desktops, mobilephones aroundthe globe.

d.

Problem Solution fit

CHAPTER 4

REQUIREMENT ANALYSIS

4.1Functional requirements Hardware Requirements:

Operating System	Windows, Mac,Linux
CPU (for training)	Multi CoreProcessors (i3 or above/equivalent)
GPU (for training)	NVIDIA AI Capable / Google's TPU
Web Cam	Integrated or External withFull HD Support

Software Requirements

Python	v3.9.0 or Above
Python Packages	flask, tensorflow, opencv-python, keras, numpy,pandas, virtualenv, pillow
Web Browser	Mozilla Firefox, GoogleChrome or any modern webbrowser

IBM Cloud (fortraining)	Watson Studio - Model Training & Deployment as Machine Learning Instance

CHAPTER 5

PROJECT DESIGN

a. Data Flow Diagrams

Data Flow Diagram

Flow Chart

b. **Solution & Technical Architecture**

Solution Architecture:

Solution architecture is a bridgethe gap betweenbusiness problems andtechnology solutions.Its goals are to:

i. The best tech solutionto solve existingbusiness problems.
ii. Describing the structure, characteristics, behaviour, and otheraspects of thesoftwareto projectstakeholders.
iii. Defined aboutfeatures, development phases,and solution requirements.
iv. Provided specifications according to which the solution is defined, managed,anddelivered.

Example - Solution Architecture Diagram:

c. User Stories

User Type	Func onal Require me nt(Epic)	User Story Numb er	User Story /Task	Acceptance criteria	Priority	Relea se
Cust omer (Mobi leuser)	Register withthe users information.	USN-1	As a user, I can registerfor the applica on by entering my email,passwor d, and confirming my password.	I can access my account /dashboard in theapplica on.	High	Sprint-1
Cust om er (Deaf peopl e)	To communica tewith peopleusing signs.	USN-2	As a user, I can see myapplica on and made changes in any browser and register to it.	I can loginand see my accountin the applica on anywhere atany me.	High	Sprint-1

Customer(Dumb people)	To communicate with people easily andefficiently.	USN-3	As a user, I can see my applica on and made changes in any browsers and register to it.	I can loginand see myaccount in the applica on anywhere.	High	Sprint-1
Customer (Normal people)	User needs to communicate with specially abled people.	USN-4	As a user, I can register for the applicaon by entering my email,password, and confirma on ismade.	I can login and seemy accoun t.	Medium	Sprint-2
Customer (Learner ofSign language)	User needs to be awareand learn aboutsign languag e.	USN-5	As a user,I can createmy account in the applica on withmy emailand password, to get knowledge aboutsignlanguages.	I can createmy account andaccess thedashboard in theapplica on.	High	Sprint-1
Cust om er (Web user)	They want the update on the applica oncondi on.	USN-6	As a user, I can register forthe applica on by entering my email, password, andconfirming my password.To get details about real-me communica on.	I can able to use any browser to access the applica onfromanywhere, to know anything about real-mecommunica o n.	High	Sprint-1
CustomerCare Executive	They want to helppeople bysending applica on condi ons.	USN-7	As a user, I can receivea message from the administra on about condionsof applica on of real- me communica on.	I will analyse andsend SMStothe people.	High	Sprint-1

CHAPTER 6 PROJECT PLANNINGAND SCHEDULING

a. Sprint Planning And Estimation

Milestone	Functional Requirement (Epic)	Milestone Story Number	Milestone Story/Task
Milestone 1	Data Collection	M1	We'recollecting dataset for building our project and creating two folders, onefor training and another one for testing.
Milestone 2	Image Preprocessing	M2	Importing imagedata generator libraries and applying imagedata generator functionality totrainthe test set.
Milestone 3	Building Model	M3	Importing themodel building libraries, Initializing the model,AddingConvolution layers, Adding the Pooling layers,Adding the Flattenlayers, AddingDenselayers, Compiling themodel Fit and Save the model.

Milestone 4	Testing Model	M4	Import the packages first.Then we savethe model and Loadthe test image, preprocess it and predict it.
Milestone 5	Application Layer	M5	Build theflask application andthe HTMLpages.
Milestone 6	TrainConversation Engine	M6	Register forIBMCloud and train Image Classification Model
Milestone 7	Final Result	M7	To ensureall the activities and resulting the finaloutput.

MILESTONE ACTIVITYPLAN

SPRINT PLANING

Sprint	Func onal Requirement(Epic)	User StoryNumber	User Story/Task	Story Points	Priority

Sprint – 1	Dataset Collection	USN – 1	Collect Dataset forbuildi ng model	9	High

Sprint – 1	Image Preprocessing	USN – 2	Perform preprocessing techniques on thedataset	8	Medium
Sprint – 2	Model Building	USN – 3	Import the required libraries, add the necessary layersand compile the model	10	High
Sprint – 2		USN – 4	Training the imageclassifi ca on modelusing CNN	7	Medium
Sprint – 3	Training andTesting the Model	USN – 5	Training the modeland tes ng the model’sperformance	9	High
Sprint – 4	Applica on Developme nt	USN – 6	Conver ng the input gesture image into English Alphabets	8	Medium

b. **Sprint Delivery Schedule**

Sprint	Total StoryPoints	Duration	Sprint StartDate	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint – 1	17	6 Days	24 October, 2022	29 October, 2022	17	29 Oct 2022
Sprint – 2	17	6 Days	31 October, 2022	05 November, 2022	17	05 Nov 2022
Sprint – 3	9	6 Days	07 November,2022	12 November, 2022	9	12 Nov 2022
Sprint – 4	5	6 Days	14 November,2022	19 November, 2022	8	19 Nov 2022

Velocity

Average Velocity= Velocity

Sprint Duration

i. Average Velocity → AV

ii. Velocity → Points per sprint

iii. Sprint Duration → Number of days per sprint

1. Sprint – 1: AV = 17÷6 = 2.83

2. Sprint – 2: AV = 17÷6 = 2.83 '

3. Sprint – 3: AV = 9÷6 = 1.5

4. Sprint – 4: AV = 5÷6 = 0.83

C. Report From Jira

BURNDOWN CHART

CHAPTER 7

CODING AND EXECUTION

a. Feature 1

Theproposed system consistsof two features front end and backend.The frontend is designed using HTML and CSS. The first feature is a webpage whenever a user wants totranslate the sign language to English, they can go to the webpage it has start button. On pressing the start button, it will turn on the camera for live translation. Once the camera is turned on, we can start translating.

Coding:

```
<!DOCTYPE html>

<html>

<head>

<title>Real Time Communication</title>
```

```
 <style>  body{  background-image: linear-
gradient(to bottom right,blue, black); background-
repeat: no-repeat; background- attachment: fixed;
 }
 h1,h2,a,p{  color:white;
}    </style>

 </head>
 <body>
 <div class="title">
 <h1><center>
REAL-TIME COMMUNICATION SYSTEM POWERED BY AI
FOR SPECIALLY ABLED</center></h1>
</div>
<center><img src="../static/img/img.png" width="300" height="300"></center>
<div>
<center><h2>Show theseGestures to get the Alphabet</h2></center>
</div>
<div>
<center><a href="{{ url_for('predict') }}">CLICK HERE TO SHOW
YOURGESTURES</a></center>
</div>
```

 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-muteand a normal personhas always been a challenging task. It is very difficult for mute people to convey theirmessage to normal people. Since normal people are not trained on hand sign language.In emergency times conveying their message is very difficult.

The project aims to develop a system that converts the sign language into a alphabet in the desired language to convey a message to normal people. We are making use of a convolution neural network to create a model that is trained on differenthand 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 isgiven as output.

</body>

</html>

b. Feature 2

The second feature of the proposed system is backend.The backend is designedusingpython with the packages of python like flask, tensorflow, opencv-python, keras,numpy,pandas, virtualenv, pillowand Machine learningtechnology and trainedwithdatasets. Once the camera

is turnedon the system detects and identify the sign languageand translate it to English by matching the live action with the trained dataset.

Coding: fromflask import Flask,render_template,request import cv2

from keras.models import load_model

import numpy as npfrom gtts import gTTS

import os from keras.preprocessing

import image from skimage.transform

import resize from playsound import

playsound app = Flask(_name_)

model=load_model("aslpng1.h5")

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

@app.route('/',

methods=['GET'])

defindex():

return

render_template('i

ndex.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 orH
is None:
                        (H, W) = frame.shape[:2]
output = frame.copy()
                        # r =
                    cv2.selectROI("Slec
                    t", 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] cleaningup...")
vs.release()
cv2.destroyAllWindows()
returnrender_template("index.html
```

")if

name == '_main_':

app.run(debug=True)

C

H

AP

T

ER

8T

ES

TI

N

G

```
# Importing Libarriesfrom
tensorflow.keras.modelsimport
load_model from
tensorflow.keras.preprocessing
import image import numpy as
np import cv2 # loadingmodel

model =
load_model('aslpng1.
h5') from
skimage.transform
import resize def
detect(frame):
    img                =
resize(frame,    (64,
64,    3))    img    =
np.expand_dims(i
mg,   axis   =   0)   if
np.max(img)   >   1:
img   =   img/255.0
prediction=
```

```
model.predict(img)

print(prediction)

return prediction

frame =

cv2.imread(r"D:\Re

al-time

Communication

System  for

speciallyabled\Data

set\test_set\A\16.

png")data =

detect(frame)

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

index[np.argmax(data)]#

Importing Librariesimportcv2

import numpy as np from

tensorflow.keras.models import

load_model from

tensorflow.keras.preprocessing

importimage

# Loading Model

model =
```

```
load_model("aslpn
g1.h5") video =
cv2.VideoCapture(
0)
index = ['A','B','C','D','E','F','G','H','I'] while
True:
    success, frame = video.read()
cv2.imwrite('frame.jpg', frame)img =
image.load_img('frame.jpg', target_size =
(64, 64))      x = image.img_to_array(img)     x = cv2.cvtColor(x,
cv2.COLOR_BGR2HSV)      a =
x.array_to_img(x)                       cv2.imshow("")        x     =
np.expand_dims(x,   axis     =    0)                        pred    =
np.argmax(model.predict(x), axis = 1)  y = pred[0]      copy
= frame.copy()     cv2.rectangle(copy, (320,100), (620,
400), (255, 0, 0), 5)       cv2.putText(frame, "ThePredicted Alphabet : " +
str(index[y]), (100, 100),  cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)
cv2.imshow('frame',
frame)                    if
```

```
cv2.waitKey(1) & 0xFF ==

ord('q'):

breakvideo.release()

cv2.destroyAllWindows()
```

CHAPTER 9

RESULT

a. **Performance Metrics**

CHAP

TER 10

ADVANTAGE

AND

DISADVANTAGE

ADVANTAGE:

i. Communication is the key in this society people with disability tends suffer butthe proposed system provides a solution to them.

ii. Makes the translation of sign languageto English easy.

iii. It can identify and translate the live and moving images.

iv. The proposedsystem ensures the easy translation of sign languageto English.

v. Even the people with lack of sign language can use the proposed system easily.

vi. This does not requirehigh-end device to use it.

vii. Can be used on almost all operating systems

and browses. ● Does not requireprior

programming knowledge t use the system

viii. The proposed systemis user friendly.

ix. Makes the life of the personwith disability easy.

DISADVANTAGE:

x. The proposed system is not a two-way translation system.

xi. There is chance for wrong translation.

xii. Since it is a webpage-based system,it does require internet connectivity whichcan be inconvenient at times.

xiii. It would have been convenient if it is application based.

C

H

AP

T

ER

11

C

O

N

CL

US

IO

N

Sign language is a useful tool for facilitating communication between deaf and hearing people. Because it allows for two-way communication, the system aims to bridge the communication gap between deaf people and the rest of society. The proposed methodology translates language into Englishalphabets that are understandable to humans.This system sends hand gesturesto the model, who recognizes them and displaysthe equivalent Alphabet on the screen. Deaf-mute people can use their hands to perform sign language, which will then be converted into alphabets, thanks to this project.

CHAPTER 12

FUTURE SCOPE

In the future to take the project to the next level two way communication system such as sign language to english and english to sign language is beign under the planning phase.The application version of the web page for both ios and androidis also in planning process for the future development.Research to improve the accuracy of the system is under progress.

C

H

AP

T

ER

13

AP

PE

N

DI

X

SOURCE CODE:

HTML:

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Real Time Communication</title>
<style>  body{  background-image: linear-
gradient(to bottom right,blue, black); background-
repeat: no-repeat; background- attachment: fixed;
}
h1,h2,a,p{  color:white;
}
</style>
</head>
<body>
<div class="title">
<h1><center>
REAL-TIME COMMUNICATION
SYSTEM POWERED BY AIFOR
SPECIALLY ABLED</center></h1>
</div>
<center><img src="../static/img/img.png" width="300" height="300"></center>
<div>
<center><h2>Show these Gestures to get the Alphabet</h2></center>
```

CLICK HERE TO SHOW YOURGESTURES

 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-muteand a normal personhas always been a challenging task. It is very difficult for mute people to convey theirmessage to normal people. Since normal people are not trained on hand sign language.In emergency times conveying their message is very difficult.

The project aims to develop a system that converts the sign language into a alphabet in the desired language to convey a message to normal people. We are making use of a convolution neural network to create a model that is trained on differenthand 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 isgiven as output.

</body>

</html> PYTHON:from flask import

Flask,render_template,request

import cv2 from keras.models

```
import load_model import
numpy asnp from gtts import
gTTS import os from
keras.preprocessing import
image from skimage.transform
import resize from playsound
import playsound app =
Flask(_name_)
model=load_model("aslpng1.h5
")
vals = ['A', 'B','C','D','E','F','G','H','I']
@app.route('/',
methods=['GET'])
defindex():
        return
render_template('i
ndex.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 orH

is None:

                        (H, W) = frame.shape[:2]

output = frame.copy()

                        # r =

                    cv2.selectROI("Slec

                    t", 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] cleaningup...")
vs.release()cv2.destroyAllWindows()
return render_template("index.html") if
```

name == '_main_':

app.run(debug=True)

TRAINNING CODE: # Importing Libraries from

tensorflow.keras.preprocessing.imageimport ImageDataGenerator #

Image Augmentation train_datagen =

ImageDataGenerator(rescale =1./255, shear_range

= 0.2, zoom_range =

0.2, horizontal_flip =

True) test_datagen =

ImageDataGenerator(re

scale = 1./255)

Loading train and test set

X_train = train_datagen.flow_from_directory(r"D:\Real-time Communication System for speciallyabled\Dataset\training_set", target_size = (64, 64), batch_size = 32, class_mode = 'categorical')

X_test = test_datagen.flow_from_directory(r"D:\Real-time Communication System for speciallyabled\Dataset\training_set", target_size = (64, 64), batch_size = 32, class_mode = 'categorical')

checkingindices

X_train.class_indices# Importing Libraries from

tensorflow.keras.models importSequential from

```
tensorflow.keras.layers import Dense from
tensorflow.keras.layers importConvolution2D, MaxPooling2D,
Flatten
# Initializing the Model model = Sequential() # Adding
Convolution Layer model.add(Convolution2D((32), (3,3),
input_shape = (64, 64, 3), activation ='relu'))
# AddingPooling Layer  model.add(MaxPooling2D(pool_size
= (2, 2))) # Adding Flatten Layer
model.add(Flatten()) #Adding Hidden
Layer model.add(Dense(units = 512,
kernel_initializer =
'random_uniform', activation = 'relu'))
# Adding Output Layer model.add(Dense(units = 9, kernel_initializer =
'random_uniform',activation = 'softmax')) # Compile the model
model.compile(loss = 'categorical_crossentropy', optimizer = 'adam',
metrics = ['accuracy']) # Fiiting the model model.fit_generator(X_train,
steps_per_epoch = 24,epochs = 10, validation_data = X_test,
validation_steps = 40) # Saving the model model.save('aslpng1.h5')
```

TESTING CODE:

```
# Importing Libarriesfrom
tensorflow.keras.modelsimport
load_model from
tensorflow.keras.preprocessing
```

```
import image import numpy as np
import cv2 # loading model

model =
load_model('aslpng1.h5') from
skimage.transform import resize
def detect(frame):
   img =
resize(frame, (64,
64, 3)) img =
np.expand_dims(i
mg, axis = 0)if
np.max(img) > 1:
img = img/255.0
prediction =
model.predict(img)     print(prediction)

   return prediction frame = cv2.imread(r"D:\Real-time
Communication Systemfor
speciallyabled\Dataset\test_set\A\16.png")data =
detect(frame)
index = ['A','B','C','D','E','F','G','H','I']
index[np.argmax(data)]#
Importing Librariesimportcv2
```

```
import numpy as np from
tensorflow.keras.models import
load_model from
tensorflow.keras.preprocessing
import image
# Loading Modelmodel =
load_model("aslpng1.h5")video =
cv2.VideoCapture(0)

index = ['A','B','C','D','E','F','G','H','I'] while
True: success,
frame = video.read()
cv2.imwrite('frame.jpg', frame)
img = image.load_img('frame.jpg',
target_size = (64, 64)) x =
image.img_to_array(img) x =
cv2.cvtColor(x, cv2.COLOR_BGR2HSV)
a = x.array_to_img(x)

cv2.imshow("") x = np.expand_dims(x,
axis = 0) pred =
np.argmax(model.predict(x), axis = 1)
y= pred[0]
```

```
copy

= frame.copy()

cv2.rectangle(copy,

(320, 100), (620,

400), (255, 0, 0), 5)

    cv2.putText(frame,"The Predicted Alphabet: " + str(index[y]), (100,100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)

cv2.imshow('frame',

frame)                    if

cv2.waitKey(1) & 0xFF ==

ord('q'):

        break



video.release()

cv2.destroyAllWindows()
```

GITHUBGITHUBLINK:

https://github.com/IBM-EPBL/IBM-Project-50772-

1660923765**DEMO LINK:**

https://drive.google.com/file/d/1WIKXemxc191hgiypb9pOQUXTMpNyYxnl/view?usp=drivesdk