REAL - TIME COMMUNICATION SYSTEM PROWERED BY AI FOR SPECIALLY ABLED

ABSTRACT

This project aims to aid the deaf-mute by creation of a new system that helps convert sign language to text and speech for easier communication with audience. The system consists of a gesture recognizes hand-glove which convert gestures into electrical signals using flex scensors.These electrical signals are then proceed using an Arduino microcontroller and a python based backene for text-to-speech conversation.The glove includes two modes of operation phrase fetch mode and letter fetch mode. The phrase fetch mode speaks out words at once,while the letter fetch mode speaks out individual letters.

1.INTRODUCTION

Gesture is a non- verbal means of communication. It refers to expressing an idea using position, orientation or movement of a body part. Gesture recognition is the mathematical intrepretation of orientation or motion of human body by a computational system. In this project,the words expressed by hand gestures by the speech and hearing impaired are converted into verbal means of communication.The translated output is displayed on a screen and speaker.

1.1 PROJECT OVERVIEW

Sign language is the well- structured code which uses hand gestures instead of sound to convey meaning simultaneously combining hand shapes, orientation and movement of the hands . Communicative hand glove is an electronic device that can translate sign language into speech and text in order to make the communication possibles between the deaf and mute with the general people.

This technology has been used in a variety of application areas which demands accurate intrepretation of sign languages. In this project the words/ letters conveyed by the disabled person are displayed on a screen

1.2 PROPOSE

There are many different types of disabilities and there are also many different ways in which people may use AI. Some people may use it as an assissstant for completing daily tasks. Some may use it as a visual aid or it read text aloud. Others may need to relay an AI is setting the pace of the future and helping people in need.The use of artificial intelligence is a boon for specially abled people. Technology had opened up new oppourtunities and created jobs where none had existed before, such as speech to text software that helped one woman find her voice .

AI technology impact lives facial recognition and predictive texting tools allow some individuals who have difficulty speaking to communicate more easily on the phone or through socail media channels like Facebook, Messenger automated system can detect when someone fall or trips unexpectedly so emergency service will be notified quickly.

2.LITREATURE SURVEY

This system consists mainly two modules the first module is Indian sign Language gestures from real-time video and mapping it with human-understandable speech.

According, the second module is the natural language as input and card with equivalent Indian Sign language animated gestures. This paper presents design and implemented of real-time sign language recognition system to 26 gestures from the Indian sign languages.In this system edges detection algorithm is used to recognize the input characters image gray scale and recognition of the edges of the hand gestures.

2.1 EXISTING PROPOSED

Deaf and dumb people are usually face some people on normal communication with other people in society. It has been observed that they sometimes find it difficulties to interact with normal people with their gestures. Because people with hearing problems or deaf people cannot speak like normal people, they have to depend on a kind of visual communication in most cases.

To overcome these problems, we have proposed a system that uses cameras to capture and convert the videos of hand gestures from dumb people who turn into

speech for hand gestures.

2.2 REFERENCE

1. Prof.P.G Ahire Tirkey T.A Jawake P.B Warale
" Two way communication between deaf and dumb people and normal people"

2. Shreyashi Narayana Swami
"Sign language recoginition system to deaf- dumb people using pca"

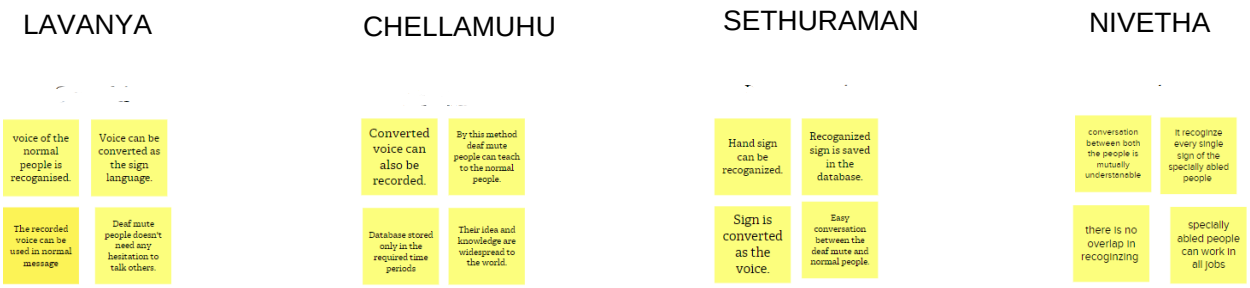
3. Anithkumar Shnide, Ramesh Kagaikar,
"Sign language to text and vice versa recoganition using computer vision in Marathi"

4. Setiwardhana, Rizky,Yuniar Hakkun
"sign language learning based on android for deaf and speech impaired people"

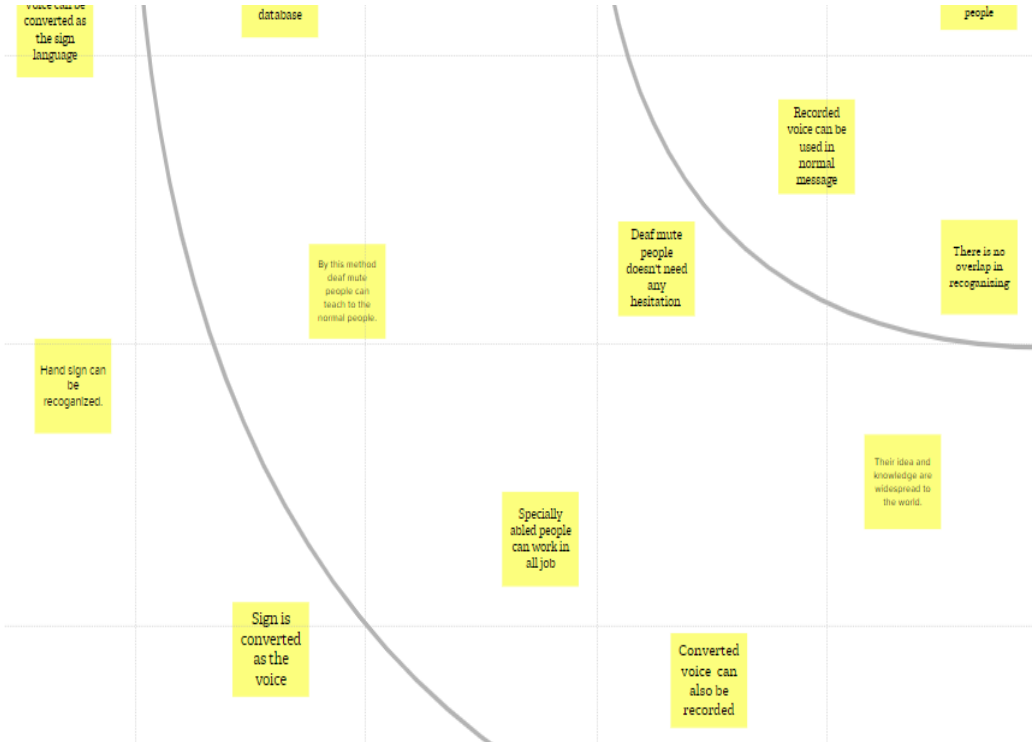
5. M.Ebrahmin Al-Ahdal & Noortawati Md Tahir,
"Review in sign language recognition systems.

3 . IDEATION & PROPOSED SOLUTION

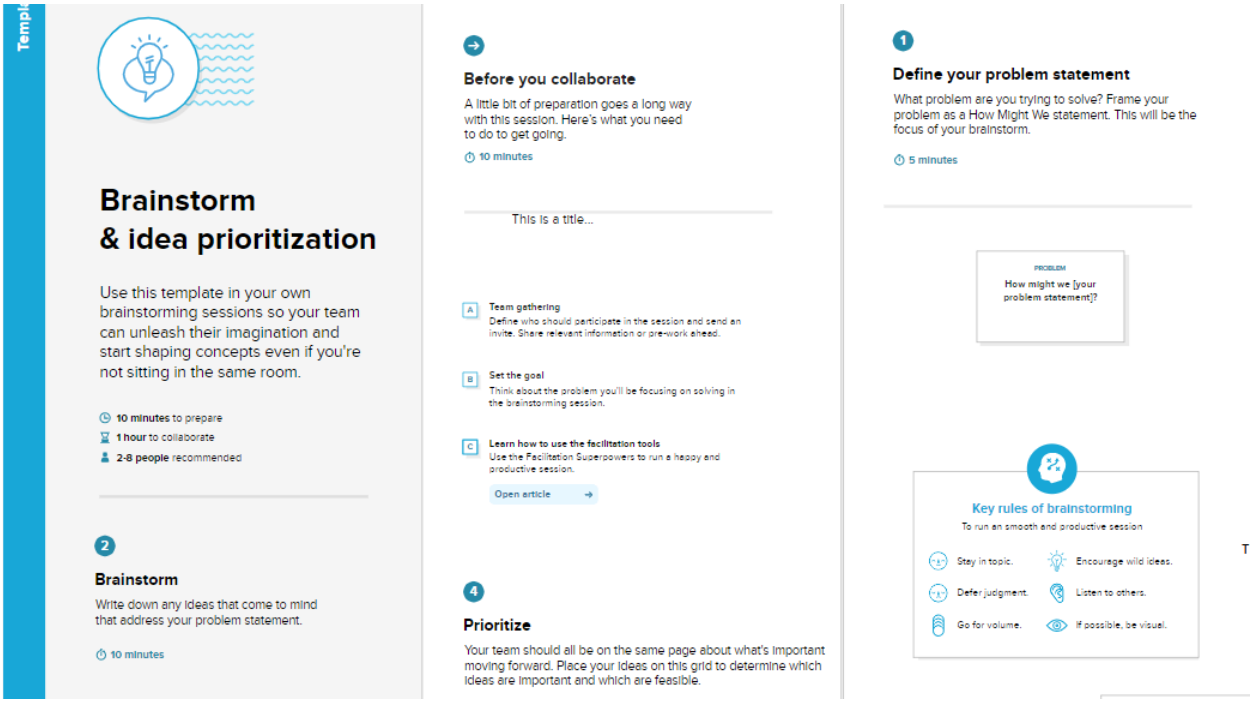
3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTROMING



Brainstroming



3.3 PROPOSED SOLUTION

Project Design Phase-I
Proposed Solution Template

Date	11 October 2022
Team ID	PNT2022TMID48010
Project Name	Real - Time Communication System Powered by AI for Specially Abled
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Conversation between deaf-mute people and normal people is a very challenging task. It is most difficult to communicate during the emergency period.
2.	Idea / Solution description	• To design and implement a system using artificial intelligence, Data mining and image processing concepts to take input as hand gestures.

3.4 PROBLEM SOLUTION FIT

Project Design Phase-1
Solution Fit Template

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

Team ID: PNT2022TMID48010

1. CUSTOMER SEGMENT(S) CS
Who is your customer?
i.e. working parents of 0-5 y.o. kids

Deaf-Mute people had difficulties to communicate and express their ideas to the normal people through their sign language.

5. CUSTOMER CONSTRAINTS CC
What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

The main constraints would include,
1. The specially abled people wants to share their Knowledge with their sign language.
2. Also the specially abled people like to know the reply form normal people.

6. AVAILABLE SOLUTIONS AS
Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

1. The recoganisation of sign language from hand geastures, face expression should be recoganised and it will be stored in the database.
2. The stored sign language can be converted into voice for the normal people.
3. The voice of normal people can stored and it should be converted as the sign geastures,

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	User Must register through their Gmail
FR-2	User Communication	Communication can be done through the PC or laptop
FR-3	User communication	The user Has to communicate in Front of the Camera
FR-4	User Requirement	Option should be shown • Hand sign to voice conversion • Voice conversion to hand sign
FR-5	Reporting	If any issue found in the app automatically it will be alert to the developer.

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 camera captures all the expressions including facial expressions and hand gestures which can be easily recoganised by all age groups.
NFR-2	**Security**	The system is more secure, the data stored in the database are confidentialy.
NFR-3	**Reliability**	The system is very liable, it can last for long amounts of time if well maintained.

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

Flow:

1. Collecting the key points from media pipe holistic and also collect a bundle of data from key points.
2. Then build a LSTM model and train with stored data.
3. The number of epoches for the model is determined by us, if we increases the number of epoches the accuracy increase but time take to run the model also increase and overfitting of model can happen, for gestures recoganise.

4. After completion of training we can use this model for real time hand geastures detection and simultaneously convert the gesture to speech using opencv.

5.2 SOLUTION & TECHNICAL ARCHITECTURE

1	User	**Deaf and dumb people intrested and willing to communicate efficiency**	**AI Technology**
2	Flask	**The components of flask user interface allow one to interact with others**	**Can be encoded using the cloud technology**
3	**Image**	**The prototype of the application is trained on the dataset**	**AI Technology**
4	**Training**	**SVM is run on the trainig and dataset to absstract the attribute from the images**	**Natural Language Processing**

5.3 USER STORIES

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Developer	Model Building	USN-1	Collect Dataset		High	Sprint-1
		USN-2	By use of Media Pipe Holisitic can collecting key points.		High	Sprint-1
		USN-3	Training a Model Using LSTM from key Points		High	Sprint-2
		USN-4	Convertion of text to Speech using the googleApi		Medium	Sprint-2
		USN-5	By using flask app model is integrated.		High	Sprint-3
Customer (Web user)	Communication	USN-1	Communication can be done through in front of camera		High	Sprint-1
		USN-2	Speech and text are delivered by web interface		High	Sprint -1

6.PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	04 November 2022
Team ID	PNT2022TMID48010
Project Name	Project - Real-Time Communication System Powered by AI for Specially Abled
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password or through Gail, Facebook.	20	High	LAVANYA CHELLAMUTHU SETHURAMAN NIVETHA
Sprint-2	Data Input	USN-2	As a user, I will be giving the input via Camera as sign language or via speech.	20	High	LAVANYA CHELLAMUTHU SETHURAMAN NIVETHA
Sprint-3	Data Verification	USN-3	Once the user gives the data input via sign languages or speech it verifies with the database.	20	High	LAVANYA CHELLAMUTHU SETHURAMAN NIVETHA
Sprint-4	Final Delivery	USN-4	Verifies with the data set and converts the input to text messages.	20	High	LAVANYA CHELLAMUTHU SETHURAMAN NIVETHA

6.2 SPRINT DELIVERY SCHEDULE

Deliverable

Team id: PNT2022TMID48010

Paper 1: Caring of Disabilities Deaf Mute Patient with Talking Devices Application Based on Mobile

Publication year: 2018

Author: Nian Afrian Nuari

Journal Name: International Journal of Engineering & Technology

The role of nurses in children with disable is to help in communicating so they can interact with others. Deaf mute requires health care and information on health education. This research focused on an application that facilitates Disabilities Deaf Mute Patient to communicate with other by using a mobile phone. This application based on mobile user by typing letters and it will automatically change into the form of voice (text to speech) that have been arranged into a word so can understood by the other person who use the application. This research helps nurses to communicate by using talking tools application. Development of systems using mobile technology by using Java programming language and talking tool. Results of this study

8.TESTING

8.1 TEST CASES

Import The Packages And Load The Saved Model

Duration: 0.5 Hrs

Skill Tags:

As a first step to start prediction we import packages that are used for loading the model and used to expand the dimension of the image. We use the Keras package to load the model which was saved when we built the model.

```
from keras.models import load_model
import numpy as np
import cv2
```

```
model=load_model('aslpng1.h5')
```

8.2 USER ACCEPTANCE TESTING

Load The Test Image, Pre-Process It And Predict

Duration: 0.2 Hrs

Skill Tags:

Pre-processing the image includes converting the image to the array and resizing according to the model. Give the pre-processed image to the model to know to which class your model belongs to.

```
from skimage.transform import resize
def detect(frame):
        img = resize(frame,(64,64,1))
        img = np.expand_dims(img,axis=0)
        if(np.max(img)>1):
            img = img/255.0
        prediction = model.predict(img)
        print(prediction)
        prediction = model.predict_classes(img)
        print(prediction)
```

```
frame=cv2.imread(r"G:\Gayatri Files\Smartbridge\Nidhi\Conversation Engine for Deaf and Dumb\Dataset\test_set\G\1.png")
data = detect(frame)
```

```
[[6.0201724e-13 7.6744452e-18 1.7007801e-10 7.7269103e-14 2.9694178e-15
  8.9405344e-16 9.9999082e-01 9.1214142e-06 3.0555274e-17]]
[6]
```

9.RESULTS

9.1 PERFORMANCE MATRICES

OpenKore source code documentation

Main website

Table of contents

Artificial intelligence

How the AI subsystem is designed

The AI subsystem isn't really complex, but it could take a while to understand it's design.

All "intelligence" is handled inside the AI() function (right now it's one big function but we hope to split it in the future). As explained in the Main loop & initialization page, the AI() function only runs less than a fraction of a second.

Basically, the AI tells Kore to do certain things based on the current situation. I'll try to explain it with some examples.

10. ADVANTAGES & DISADVANTAGES

Advantages

1.It defines a more powerful and more useful.

2. It introduce a new and improved interface for human interaction.

3. It introduce a new technique to solve a new problems.

4. It handles the information better than humans.

Disadvantages

1. It implementation cost is very high

2. The difficulties with software development for AI implementation are that the development of

software is sllow and expensive.

3. Machine can easily lead to destruction if the implementation of machine put in the wrong hands the results are hazaradous.

4. A robot is one of the implementation with the replacing jobs .

11.CONCLUSION

The proposed communication between deaf and dumb people and ordinary people are aiming for its when bridging the communication gap between two socities several works done ealier in this area . but this paper adds in complete two stabled communication in an efficiency manner because of the system implementation are one handy mobile applications. so it really serves its need in the aspects. The above strateges prove to be efficient in terms of the time and accuracy futher improvement can be done in the implementaation of the communication with other sign languages such as American Sign Languages for different accespts throughtout the recognition of the emotion in sign language and language translation.

12.FUTURE SCOPE

* Proposed system scope be related with the eduction of dumb people . Dumb people faces many problem when normal person could not understand their languages. They were facing communication gap with normal peoples.

* For communicatuion between deaf people and a second person could not required to translste sign language of deaf people . But a mediator isvrequired to know the sign language used by deaf person.

13.APPENDIX

Source code

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

class Video(object):
        def __init__(self):
                self.video = cv2.VideoCapture(0)
                self.roi_start = (50, 150)
                self.roi_end = (250, 350)
                self.model = load_model('asl_model.h5') # Execute Local Trained Model
                # self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
                self.index=['A','B','C','D','E','F','G','H','I']
                self.y = None
        def __del__(self):
                self.video.release()
        def get_frame(self):
                ret,frame = self.video.read()
                frame = cv2.resize(frame, (640, 480))
                copy = frame.copy()
                copy = copy[150:150+200,50:50+200]
                # Prediction Start
                cv2.imwrite('image.jpg',copy)
                copy_img = image.load_img('image.jpg', target_size=(64,64))
                x = image.img_to_array(copy_img)
                x = np.expand_dims(x, axis=0)
                pred = np.argmax(self.model.predict(x), axis=1)
                self.y = pred[0]
                cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
                ret,jpg = cv2.imencode('.jpg', frame)
                return jpg.tobytes()
```

Testing the model

import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('asl_model_84_54.h5')
img=image.load_img(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set\D\2.png',
target_size=(64,64))
img

x=image.img_to_array(img)
x.ndim
3
x=np.expand_dims(x,axis=0)
x.ndim
4
pred=np.argmax(model.predict(x),axis=1)
1/1 [==============================] - 0s
88ms/step
pred
array([3], dtype=int64)
index=['A','B','C','D','E','F','G','H','I']

print(index[pred[0]])
D
Open CV

import cv2
img=cv2.imread(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set\C\2.png',1)
img1=cv2.imread(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set\B\2.png',0)
print(img.shape)
(64, 64, 3)
#
img=cv2.imread(r'C:\Users\LEGION\Desktop\Project Externship\Dataset\test_set\B\2.png',1)
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
CNN Video Analysis

import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('asl_model_84_54.h5')

```
video=cv2.VideoCapture(0)
index=[&#39;A&#39;,&#39;B&#39;,&#39;C&#39;,&#39;D&#39;,&#39;E&#39;,&#39;F&#39;,&#39;G&#39;,&#39;H&#39;,&#39;I&#39;]
while 1:
succes,frame=video.read()
cv2.imwrite(&#39;image.jpg&#39;,frame)
img=image.load_img(&#39;image.jpg&#39;,target_size=(64,64))
x=image.img_to_array(img)
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,&#39;The Predicted Alphabet is: &#39;+str(index[y]),(100,100),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),4)
cv2.imshow(&#39;image&#39;,frame)
if cv2.waitKey(1) &amp; 0xFF == ord(&#39;q&#39;):
break
video.release()
cv2.destroyAllWindows()
```

1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 15ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s

12ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
17ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 16ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s

15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 12ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s

15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
12ms/step

1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
15ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 12ms/step

1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
16ms/step
1/1Testing the model

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model(&#39;asl_model_84_54.h5&#39;)
```

img=image.load_img(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set\D\2.png',
target_size=(64,64))
img

x=image.img_to_array(img)
x.ndim
3
x=np.expand_dims(x,axis=0)
x.ndim
4
pred=np.argmax(model.predict(x),axis=1)
1/1 [==============================] - 0s
88ms/step
pred
array([3], dtype=int64)
index=['A','B','C','D','E','F','G','H','I']
print(index[pred[0]])
D
Open CV

import cv2

img=cv2.imread(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set\C\2.png',1)
img1=cv2.imread(r'E:\Projects\SmartBridge\ModelGen\Dataset\test_set\B\2.png',0)
print(img.shape)
(64, 64, 3)
#
img=cv2.imread(r'C:\Users\LEGION\Desktop\Project Externship\Dataset\test_set\B\2.png',1)
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
CNN Video Analysis

import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('asl_model_84_54.h5')
video=cv2.VideoCapture(0)
index=['A','B','C','D','E','F','G','H','I']
while 1:

succes,frame=video.read()
cv2.imwrite('image.jpg',frame)
img=image.load_img('image.jpg',target_size=(64,64))
x=image.img_to_array(img)
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 is: '+str(index[y]),(100,100),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),4)
cv2.imshow('image',frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
break
video.release()
cv2.destroyAllWindows()
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
16ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
12ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 17ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
16ms/step

1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s

12ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
16ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
16ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 12ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
16ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
16ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
12ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s

13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step

1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step

1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s

15ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
16ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s

14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
15ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step

KeyboardInterrupt Traceback (most recent call last)
e:\Projects\SmartBridge\ModelGen\Externship Project.ipynb Cell 44' in ()

7
index=['A','B','C','D','E','F','G','H&#3

9;,'I']
8 while 1:
----> 9 succes,frame=video.read()
10 cv2.imwrite('image.jpg',frame)
11
img=image.load_img('image.jpg',target_size=(
64,64)) [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
15ms/step

1/1 [==============================] - 0s
14ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
13ms/step
1/1 [==============================] - 0s
15ms/step

KeyboardInterrupt Traceback (most recent call last)
e:\Projects\SmartBridge\ModelGen\Externship
Project.ipynb Cell 44' in ()

7
index=['A','B','C','D','E','F','G','H','I']
8 while 1:
----> 9 succes,frame=video.read()
10 cv2.imwrite('image.jpg',frame)
11
img=image.load_img('image.jpg',target_size=(64,64))

GITHUB & PROJECT DEMO LINK

https://drive.google.com/file/d/1bBeNAVNPs4DSzQa90tV4kittYJkulcHa/view?usp=drivesdk