PROJECT REPORT

A GESTURE-BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

Team ID : **PNT2022TMID35486**

Team Leader – Sandhiya Ramalaxmi M(2019105568)

Team Member – Rukshana Rani R(2019105046)

Team Member – Sowndarya S (2019105580)

Team Member – Tejaswini S (2019105590)

CONTENT

1. INTRODUCTION :

1.1 Project Overview

- Interactive presentation systems use advanced Human Computer Interaction (HCI) techniques to provide a more convenient and user-friendly interface for controlling presentation displays, such as page up/down controls of x-rays and relayed images in the medical field .
- Compared with traditional mouse and the interface for human-machine keyboard control, new experience is significantly improved with these techniques.
- Hand gesture has wide- ranging applications. In this study, we apply it to an interactive presentation system to create an easy-to-understand interaction interface.
- The use of doctor-computer interaction devices in the operation room (OR) requires new modalities that support medical imaging manipulation while allowing doctors' hands to remain sterile. We are presenting "Gestix," a vision-based hand gesture capture and recognition system that interprets in real time the user's gestures for navigation and manipulation of images in an electronic medical record (EMR) database.
- "Gestix" was tested during a brain biopsy procedure. In the in vivo experiment, this interface prevented the surgeon's focus shift and change of location while achieving a rapid intuitive reaction and easy interaction.

2. LITERATURE SURVEY :

2.1 Existing Problem

- A major challenge involved is to provide Doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work.
- However the use of computer keyboards and mouse by doctors in intensive care unit(ICU) is a common mean for spreading infections.
- We suggest the use of hand gestures in medical field as an alternative to the existing interface techniques by offering maximum level of sterility.

2.2 References

- HAND GESTURE RECOGNITION: Published:2012 July Authors: Rafiqul Zaman Khan and Noor Adnan Ibraheem

- HAND GESTURE METHODS AND APPLICATIONS: Published:2021 October Authors: Lazzat Zulpukharkyzy Zholshiyeva, Tamara Kokenovna Zhukabayeva, Sherzod Turaev, Meruyert Aimambetovna Berdiyeva, Dina Tokhtasynovna Jambulova

- HAND GESTURE RECOGNITION SYSTEMS: Published:2013 May Authors: Arpita Ray Sarkar, G. Sanyal, S. Majumder

- HAND GESTURE RECOGNITION SYSTEMS: Published:2013 May Authors: Arpita Ray Sarkar, G. Sanyal, S. Majumder

- GESTURE-CONTROLLED IMAGE SYSTEM POSITIONING FOR MINIMALLY INVASIVE INTERVENTIONS: Published:2021 Authors: Benjamin Fritsch*, Thomas Hoffmann, André Mewes and Georg Rose

2.3 Problem Statement Definition

A doctor who needs a contactless image browsing tool because the risk of getting infected is high. So, a Gesturebased Tool for Sterile Browsing of Radiology Images is required. The present-day use of doctor-computer interaction devices in the operation room to browse radiology images accompanies with the risk of getting infected and frequent movement to the computer which makes it uneasy. Therefore, it requires new modalities that support medical imaging manipulation while allowing doctors' hands to remain sterile, supporting their focus of attention, and providing fast response times. By fixing the issue, doctors can ensure sterile environment in the operation theatre and experience hands-free browsing of radiology images of patients which is much easier and more convenient than the existing techniques.

3. IDEATION & PROPOSED SOLUTION :

3.1 Empathy Map Canvas

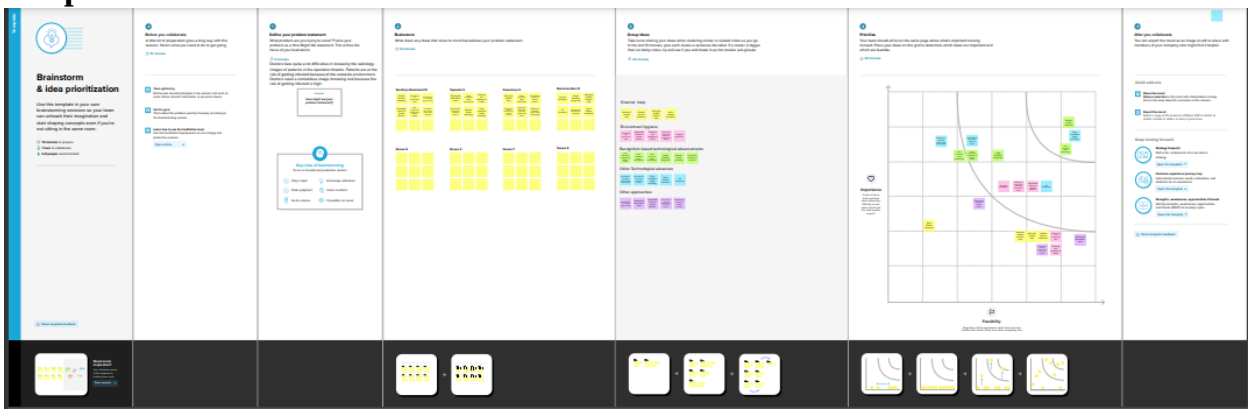
3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement.

Step-2: Brainstorm, Idea Listing and Grouping.

Step-3: Idea Prioritization.



3.3 Proposed Solution

- Gesture based Tool for sterile browsing of radiology images is the proposed solution. This ensures contactless browsing of images and thus helps the doctors to reduce movement and improve focus. Reading gestures involves the recognition of images or recordings captured by the camera. Each gesture, when identified, is translated to a specific command in the controlled application.
- Standard image processing techniques do not yield great results, gesture recognition needs to be backed by artificial intelligence to unfold its full potential. Ideally, gesture recognition should be based on a photo of a still hand showing only a single gesture against a clear background in well-lit conditions. But real-life conditions are hardly ever like that. We don't always get the comfort to use solid, clear backgrounds when presenting gestures. The role of artificial intelligence in gesture recognition is, in part, to overcome some of the main technical issues associated with proper identification of gesture
- Contributing the corporate social responsibility by providing better solutions to the healthcare and to patients. Can collaborate with diagnosis centres and hospitals and can collaborate with government for health awareness camps..

- With the addition of more radiology images data set can be expanded thus giving more efficient and effective detection of gestures and thereby improving the browsing features. The environment can be ensured to be sterile..

3.4 Problem Solution Fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Doctors and Lab technicians are the major customers. Hospitals and Health Sector are the major customer segments.

6. CUSTOMER CC

Is this product efficient and quick?
Is this product trustworthy?

5. AVAILABLE SOLUTIONS AS

There is no clear logical and feasible solution available. Cleaning the environment often is the available solution but it is time-consuming, inefficient and does not solve the problem completely.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Unsterile environment and difficulty in reaching the pointer device to browse images are the major problem.

9. PROBLEM ROOT CAUSE RC

Unclean and Unsterile environment that possesses the risk of transmitting infection to patients when doctor accesses the same pointer device in operation theatre every time. Difficulty the doctor faces to reach the pointer device while focusing on the patient.

7. BEHAVIOUR BE

Ensures the environment is clean and sterile.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

To achieve a sterile environment for the betterment of the patient and to make browsing of radiology images easier.

4. EMOTIONS: BEFORE / AFTER EM

BEFORE: Constant doubt if the environment is sterile, Inconvenience, Helplessness

AFTER: Safe and Sterile, Easy to handle and efficient

10. YOUR SOLUTION SL

A Gesture Based Tool for Sterile Browsing of Radiology Images that enables the doctor to browse images hands-free. Thus, there is no risk of the environment being unsterile, difficulty to reach the pointer device every single time and losing focus.

8. CHANNELS of BEHAVIOUR CH

8.1 **ONLINE**
Utilizing software available in the online market.

8.2 **OFFLINE**
Keeping the environment sterile.

4. REQUIREMENT ANALYSIS :

4.1 Functional Requirements

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Launching the model	Launch the trained CNN model from the cloud
FR-2	Capturing the images	After capturing the images in camera we have to upload the images in the system
FR-3	Performing gestures	After classifying, identify the correct image by the gesture and it should perform the operation
FR-4	Model rendering	After capturing the image the algorithm will start its processing task
FR-5	Sterile browsing	The sterile browsing can be performed after identifying the gestures
FR-6	Visibility of images	After completing all the processes,a user can be able to see the images

4.2 Non - Functional Requirements

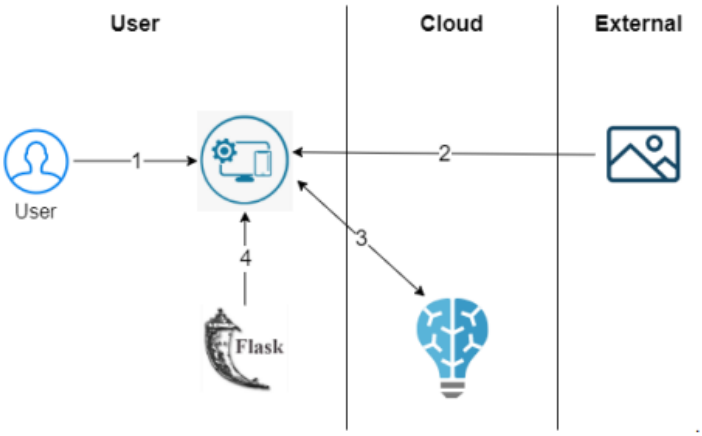
Non-functional Requirements:

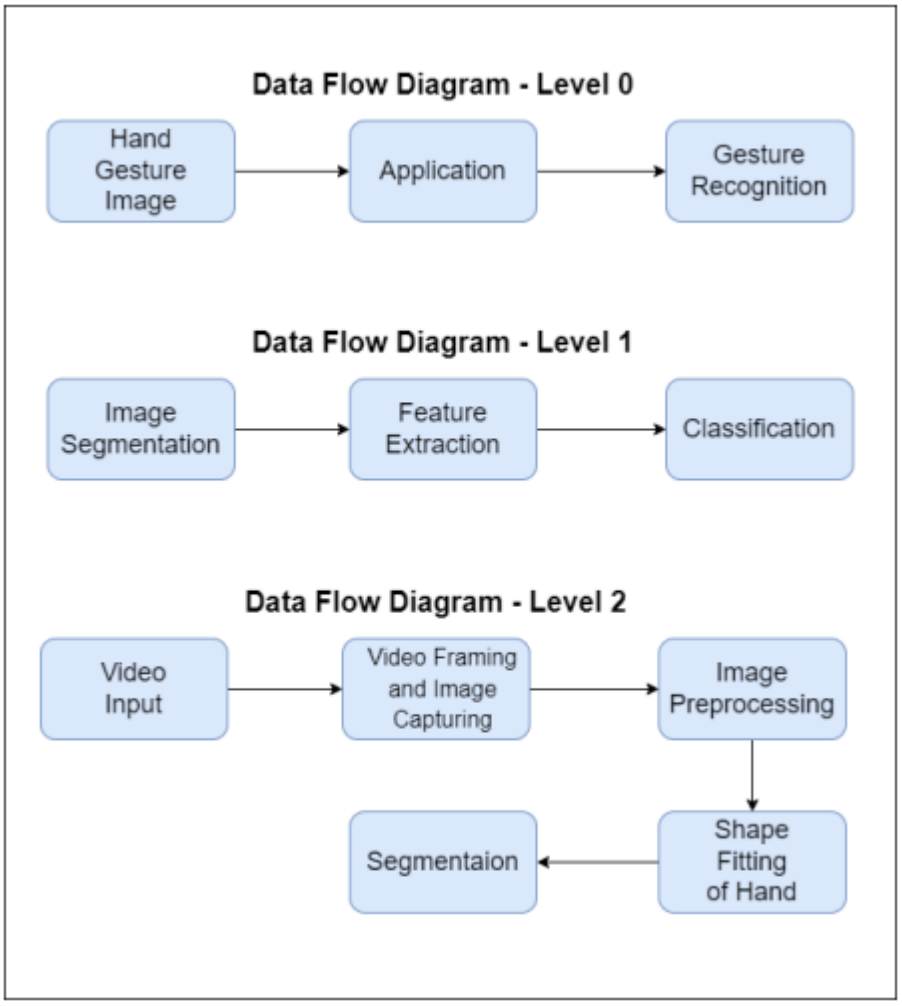
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This system helps to have the control over images without having direct contact with system which avoids the harmful rays and is ease of use
NFR-2	**Security**	This system is protected and only authorized users can access it
NFR-3	**Reliability**	After installing the application,the system will predict the gesture and performs sterile browsing
NFR-4	**Performance**	The system 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 there is an emergency
NFR-6	**Scalability**	This system allows more number of users at a time and there is no loss can be identified

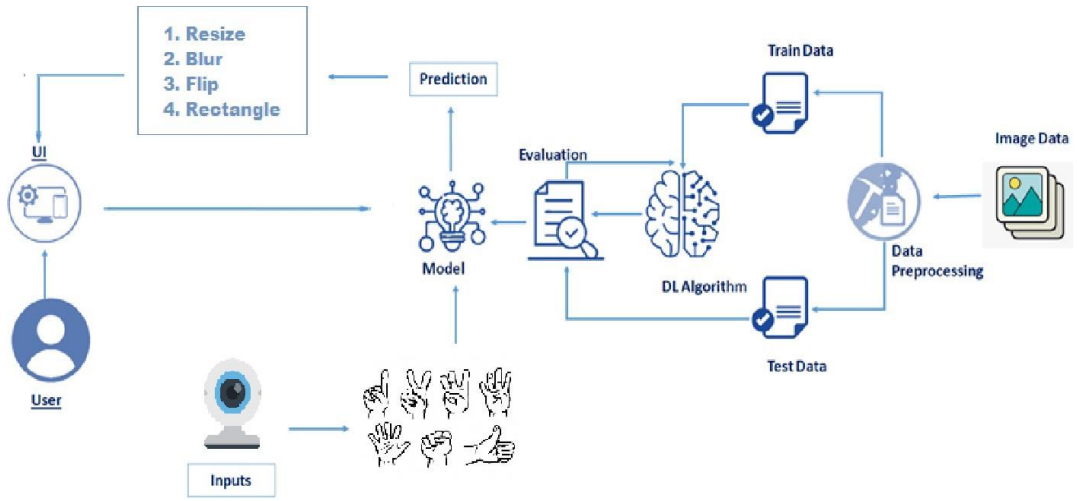
5. PROJECT DESIGN :

5.1 Data Flow Diagrams





5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Launch Web App deployed in cloud	USN-1	As a user, I can launch the web app where I can upload the images for recognition	I can upload the images for classification	High	Sprint-4
	Prediction	USN-2	As a user, I can get the predicted results from the model deployed in the cloud	I can resize the radiology image, blur the image, flip based on the hand gesture	High	Sprint-4
	Deployment of web app in the cloud	USN-3	As a user, I need the web app to be accessible all over the world	I can access the web app deployed in the IBM cloud	Medium	Sprint-3
	Deployment of AI model in the cloud	USN-4	As a user, I need the AI model to be accessible all over the world	I can access the model deployed in the IBM cloud	Medium	Sprint-3
	Model Building	USN-5	As a user, I need an AI model which could classify or recognize the hand gestures	I can get the prediction from the AI model	Medium	Sprint-1
	User Interface Building	USN-6	As a user, I need a web app for human computer interaction	I get User Interface for interaction with the model	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

To accomplish the above task, you must complete the below activities and tasks:

1.Collect the Image Data.

2.Preprocess the collected images.

3.Train the model.

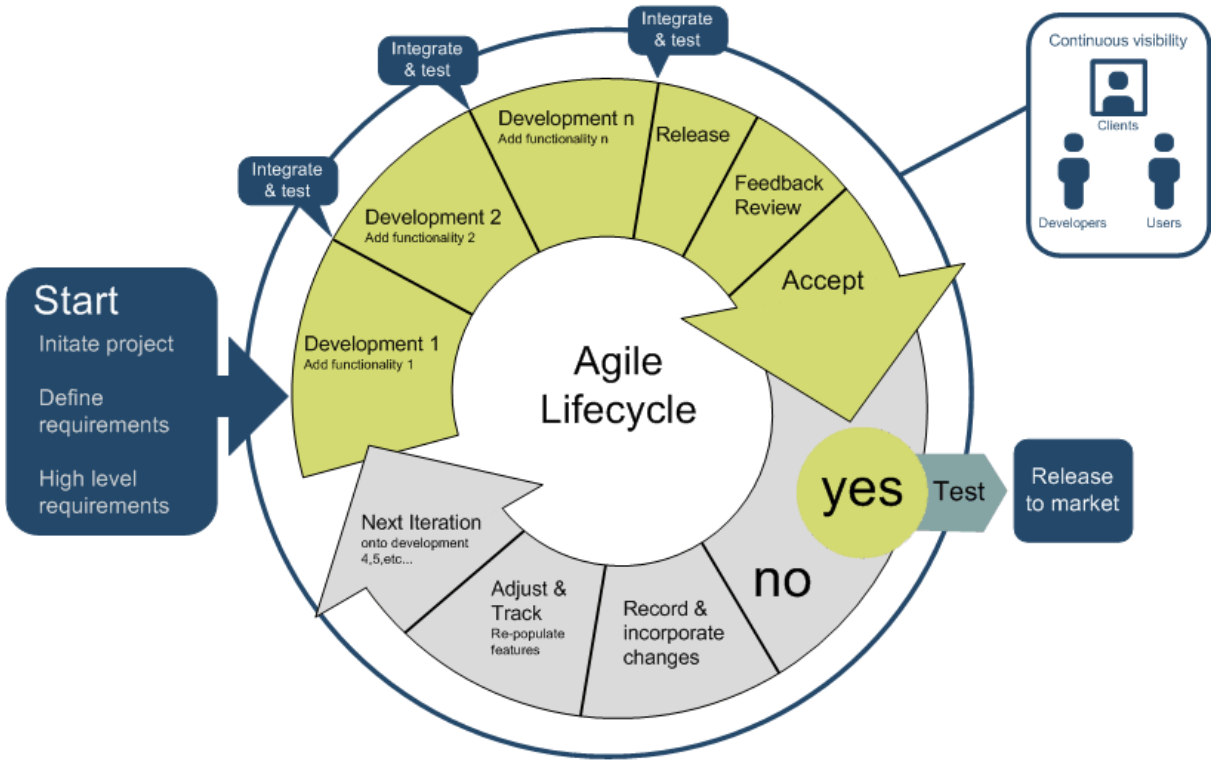
4.Test the model.

5.Model is generated (gesture.h5).

6.Apllication building using HTML & CSS.

7.Form for uploading the image for prediction.

8.Python flask for connecting Model and webpages.



6.2 Sprint Delivery Schedule

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Building IBM Watson Assistant	USN-1	Building the IBM Watson account	10	High	TM – 1 TM – 4
Sprint-1		USN-2	Collection of datasets and doing necessary prerequisites	10	High	TM – 2 TM – 3
Sprint-2	Modelling	USN-3	Creation of model so that it must predict correct hand gestures	10	Hign	TM – 1 TM – 2
Sprint-2		USN-4	Enhance the accuracy of model and adding additional features	10	Medium	TM – 3 TM – 4
Sprint-3	User Interface and Testing	USN-5	Creating a User Interface like web page	10	High	TM – 2 TM – 4
Sprint - 3		USN-6	As a user I must capture images of hand and upload it into the web portal and produce correct testing results	10	High	TM – 1 TM – 3
Sprint - 4	Model Improvision	USN-7	As a user I must receive a correct hand gesture as output	10	High	TM – 1 TM – 2
Sprint - 4		USN-8	Additional features can be made and collecting of feedback from the user to improve the quality	10	Medium	TM – 3 TM –4

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7.CODING & SOLUTIONING :

7.1.Feature 1

1:IMAGE PREPROCESSING:

Import the ImagesDataGenerator Library

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

Configure ImageDataGenerator Functionality To Trainset And Testset:

```
Image Data Agumentation

#setting parameter for Image Data agumentation to the traing data
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

Apply ImageDataGenerator Functionality To Trainset And Testset

```
Loading our data and performing data agumentation

#performing data agumentation to train data
x_train = train_datagen.flow_from_directory('data/train',target_size=(64, 64),batch_size=5,
                                            color_mode='grayscale',class_mode='categorical')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory('data/test',target_size=(64, 64),batch_size=5,
                                            color_mode='grayscale',class_mode='categorical')

Found 600 images belonging to 6 classes.
Found 30 images belonging to 6 classes.
```

2:MODEL BULIDING:

Importing The Model Building Libraries

```
Importing Neccessary Libraries

import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A Layer consists of a tensor-in tensor-out computation function
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D #Convolutional Layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator
```

Initializing The Model

```
model=Sequential()
```

Adding CNN Layers

```
# First convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Second convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the layers
classifier.add(Flatten())
```

Adding Dense Layers

```
# Adding a fully connected layer
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=6, activation='softmax')) # softmax for more than 2
```

```
classifier.summary()#summary of our model
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 62, 62, 32)        320
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten_3 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_6 (Dense)              (None, 128)               802944
_________________________________________________________________
dense_7 (Dense)              (None, 6)                 774
=================================================================
Total params: 813,286
Trainable params: 813,286
Non-trainable params: 0
_________________________________________________________________
```

```
# Adding a fully connected layer
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=6, activation='softmax')) # softmax for more than 2
```

```
classifier.summary()#summary of our model
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 62, 62, 32)        320
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten_3 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_6 (Dense)              (None, 128)               802944
_________________________________________________________________
dense_7 (Dense)              (None, 6)                 774
=================================================================
Total params: 813,286
Trainable params: 813,286
Non-trainable params: 0
_________________________________________________________________
```

Configure The Learning Process

Compiling the model

```
# Compiling the CNN
# categorical_crossentropy for more than 2
classifier.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Training Model

Fitting the model

```
classifier.fit_generator(
        generator=x_train,steps_per_epoch = len(x_train),
        epochs=20, validation_data=x_test,validation_steps = len(x_test))# No of images in test set
```

Save the Model

Saving our model

```
# Save the model
classifier.save('gesture.h5')

model_json = classifier.to_json()
with open("model-bw.json", "w") as json_file:
    json_file.write(model_json)
```

Test the Model

Predicting our results

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model = load_model("gesture.h5") #loading the model for testing
```

```
img = image.load_img(r"E:\PROJECTS\number-sign-recognition\data\test\1\1.jpg",grayscale=True,
                     target_size= (64,64))#Loading of the image
x = image.img_to_array(img)#image to array
x = np.expand_dims(x,axis = 0)#changing the shape
pred = model.predict_classes(x)#predicting the classes
pred
```

```
array([1], dtype=int64)
```

```
index=['0','1','2','3','4','5']
result=str(index[pred[0]])
result
```

```
'1'
```

1.1 Feature 2

- Let us build flask file ‘app.py’ which is a web framework written in python forserver-side scripting. Let’s see step by step procedure for building the backend application.
- App starts running when “__name__” constructor is called in main.
- render_template is used to return html file.
- “GET” method is used to take input from the user.
- “POST” method is used to display the output to the user.

1:Importing Libraries

```
from flask import Flask,render_template,request
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
import operator
import cv2 # opencv library
from tensorflow.keras.models import load_model#to load our trained model
import os
from werkzeug.utils import secure_filename
```

2:Creating our flask apllication and loading our model

```
app = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model('gesture.h5')
print("Loaded model from disk")
```

3:Routing to the html page

```
@app.route('/')# route to display the home page
def home():
    return render_template('home.html')#rendering the home page


@app.route('/intro') # routes to the intro page
def intro():
    return render_template('intro.html')#rendering the intro page

@app.route('/image1',methods=['GET','POST'])# routes to the index html
def image1():
    return render_template("index6.html")
```

```
@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def launch():
```

And the predict route is used for prediction and it contains all the codes which are used for predicting our results.

- Firstly, inside launch function we are having the following things:
    - Getting our input and storing it
    - Grab the frames from the web cam.
    - Creating ROI
    - Predicting our results
    - Showcase the results with the help ofopencv
    - Finally run the application

- Getting our input and storing it

Once the predict route is called, we will check whether the method is POST or not if is POST then we will request the image files and with the help of os

```
if request.method == 'POST':
    print("inside image")
    f = request.files['image']

    basepath = os.path.dirname(__file__)
    file_path = os.path.join(basepath, 'uploads', secure_filename(f.filename))
    f.save(file_path)
    print(file_path)
```

function we will be storing the image in the uploads folder in our local system.

- Grab the frames from the web cam

Now when we run the code a web cam will be opening to take the gesture input so we will be capturing the frames of the gesture for predicting our results.

```
cap = cv2.VideoCapture(0)
while True:
    _, frame = cap.read() #capturing the video frame values
    # Simulating mirror image
    frame = cv2.flip(frame, 1)
```

- Creating ROI

Aregion of interest (ROI)is a portion of an image that you want to filter or operate on in some way. The toolbox supports a set of ROI objects that you can use to create ROIs of many shapes, such circles, ellipses, polygons, rectangles, and hand-drawn shapes. ... A common use of an ROI is to createa binary mask image.

So, we will be creating a ROI to mask our gesture.

```
# Got this from collect-data.py
# Coordinates of the ROI
x1 = int(0.5*frame.shape[1])
y1 = 10
x2 = frame.shape[1]-10
y2 = int(0.5*frame.shape[1])
# Drawing the ROI
# The increment/decrement by 1 is to compensate for the bounding box
cv2.rectangle(frame, (x1-1, y1-1), (x2+1, y2+1), (255,0,0) ,1)
# Extracting the ROI
roi = frame[y1:y2, x1:x2]

# Resizing the ROI so it can be fed to the model for prediction
roi = cv2.resize(roi, (64, 64))
roi = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
_, test_image = cv2.threshold(roi, 120, 255, cv2.THRESH_BINARY)
cv2.imshow("test", test_image)
```

- Predicting our results

  After placing the ROI and getting the frames from the web cam now its time to predict the gesture result using the model which we trained and stored it into a variable for the further operations.

```
result = model.predict(test_image.reshape(1, 64, 64, 1))
prediction = {'ZERO': result[0][0],
              'ONE': result[0][1],
              'TWO': result[0][2],
              'THREE': result[0][3],
              'FOUR': result[0][4],
              'FIVE': result[0][5]}
# Sorting based on top prediction
prediction = sorted(prediction.items(), key=operator.itemgetter(1), reverse=True)

# Displaying the predictions
cv2.putText(frame, prediction[0][0], (10, 120), cv2.FONT_HERSHEY_PLAIN, 1, (0,255,255), 1)
```

- Showcase the results with the help of opencv

  Finally according to the result predicted with our model we will be performing certain operations like resize, blur , rotate etc.

```
#loading an image
image1=cv2.imread(file_path)
if prediction[0][0]=='ONE':

    resized = cv2.resize(image1, (200, 200))
    cv2.imshow("Fixed Resizing", resized)
    key=cv2.waitKey(3000)

    if (key & 0xFF) == ord("1"):
        cv2.destroyWindow("Fixed Resizing")

elif prediction[0][0]=='ZERO':

    cv2.rectangle(image1, (480, 170), (650, 420), (0, 0, 255), 2)
    cv2.imshow("Rectangle", image1)
    cv2.waitKey(0)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("0"):
        cv2.destroyWindow("Rectangle")

elif prediction[0][0]=='TWO':
    (h, w, d) = image1.shape
    center = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(center, -45, 1.0)
    rotated = cv2.warpAffine(image1, M, (w, h))
    cv2.imshow("OpenCV Rotation", rotated)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("2"):
        cv2.destroyWindow("OpenCV Rotation")

elif prediction[0][0]=='THREE':
    blurred = cv2.GaussianBlur(image1, (11, 11), 0)
    cv2.imshow("Blurred", blurred)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("3"):
        cv2.destroyWindow("Blurred")
else:
    continue
```

```
        interrupt = cv2.waitKey(10)
        if interrupt & 0xFF == 27: # esc key
            break


    cap.release()
    cv2.destroyAllWindows()
return render_template("home.html")
```

RUN THE APPLICATION:

At last, we will run our flask application

```
if __name__ == "__main__":
   # running the app
    app.run(debug=False)
```

Run The app in local browser

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type “python app.py” command
- Navigate to the localhost where you can view your web page

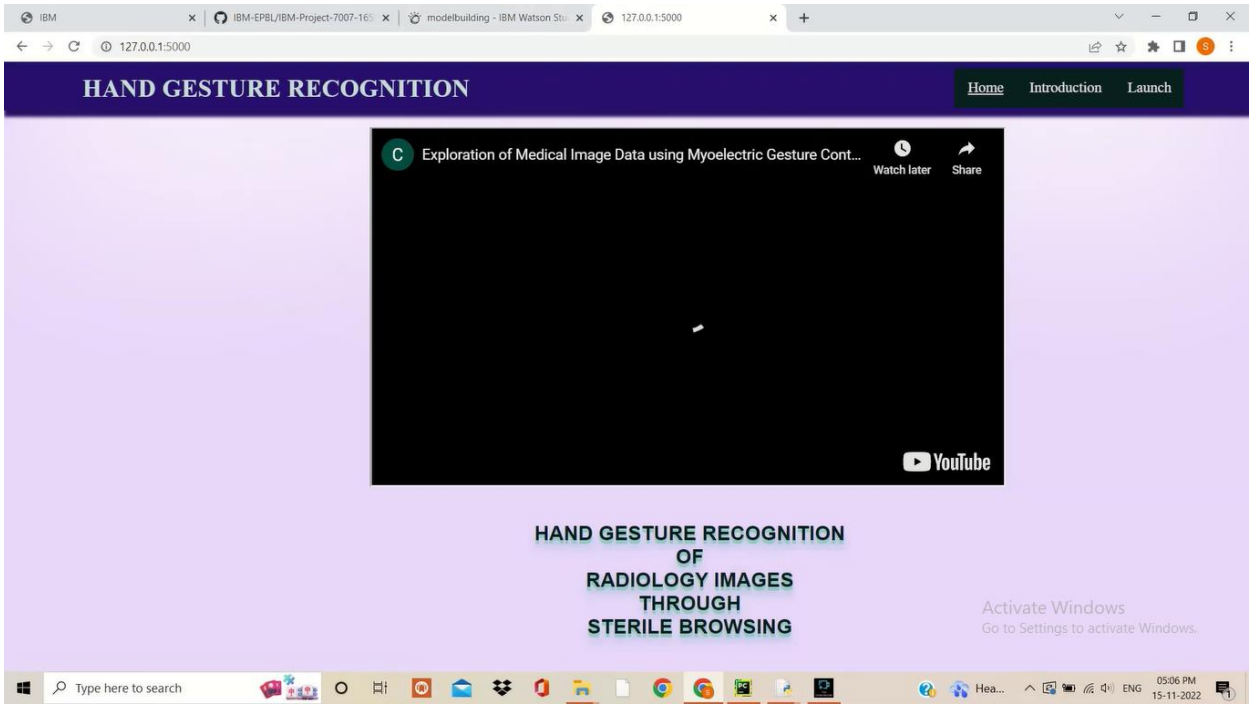
```
(base) E:\>cd E:\PROJECTS\number-sign-recognition\Flask

(base) E:\PROJECTS\number-sign-recognition\Flask>python app.py
```

Then it will run on localhost:5000

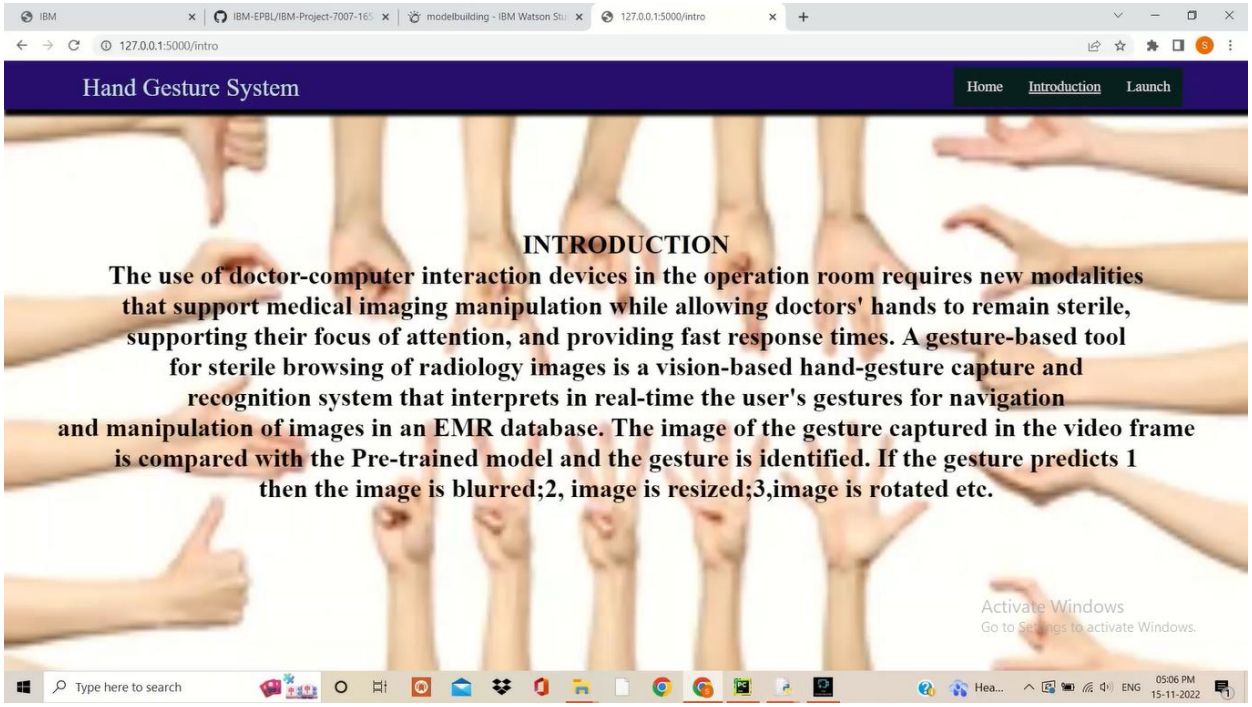
```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

8.RESULT

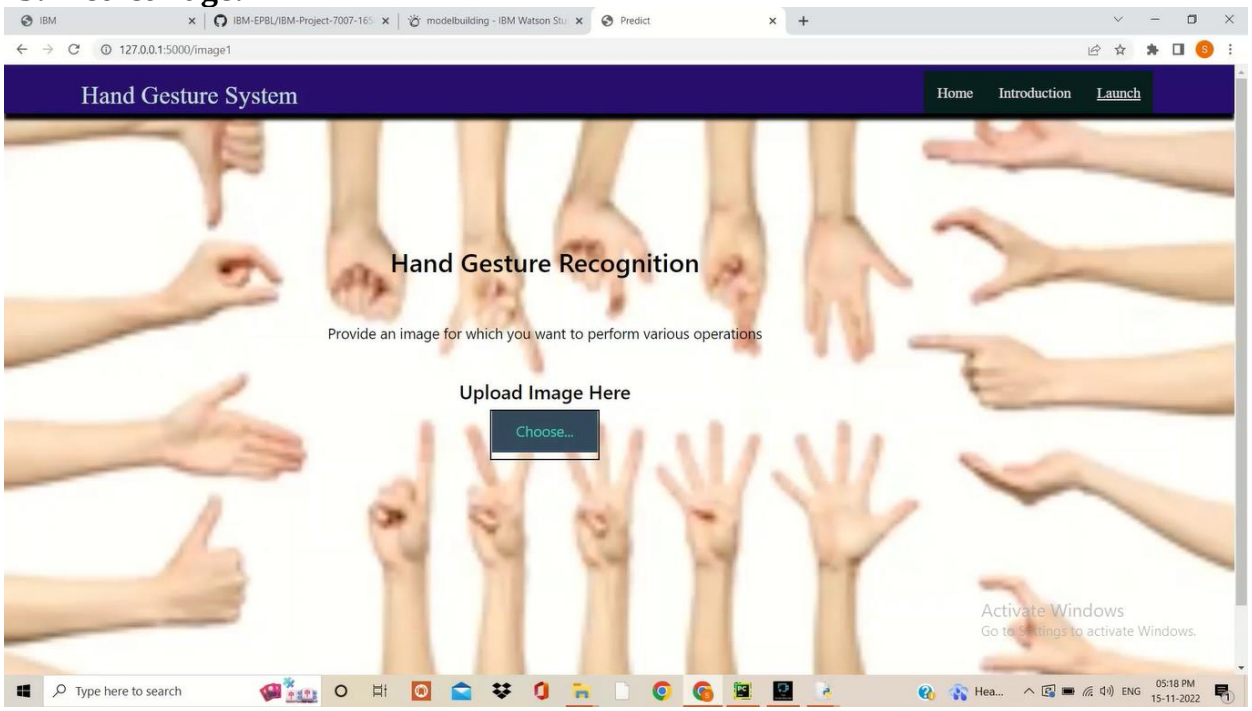
1:Home Page:



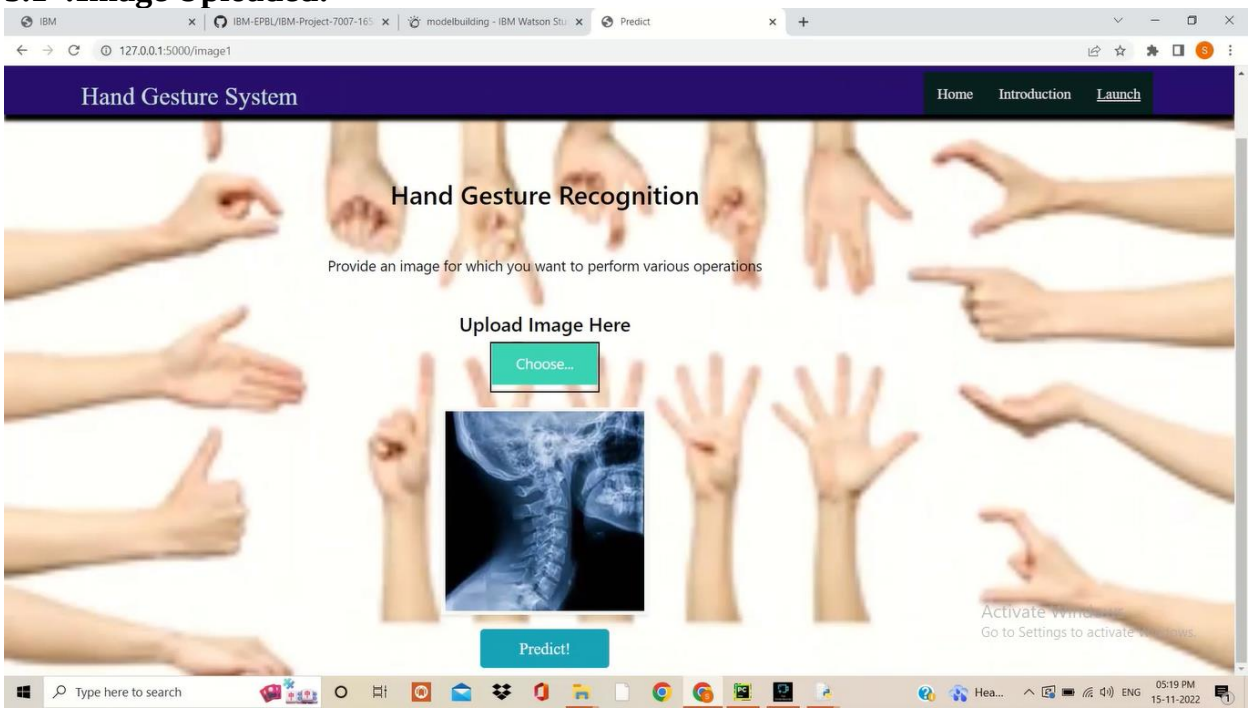
2:Introduction Page:



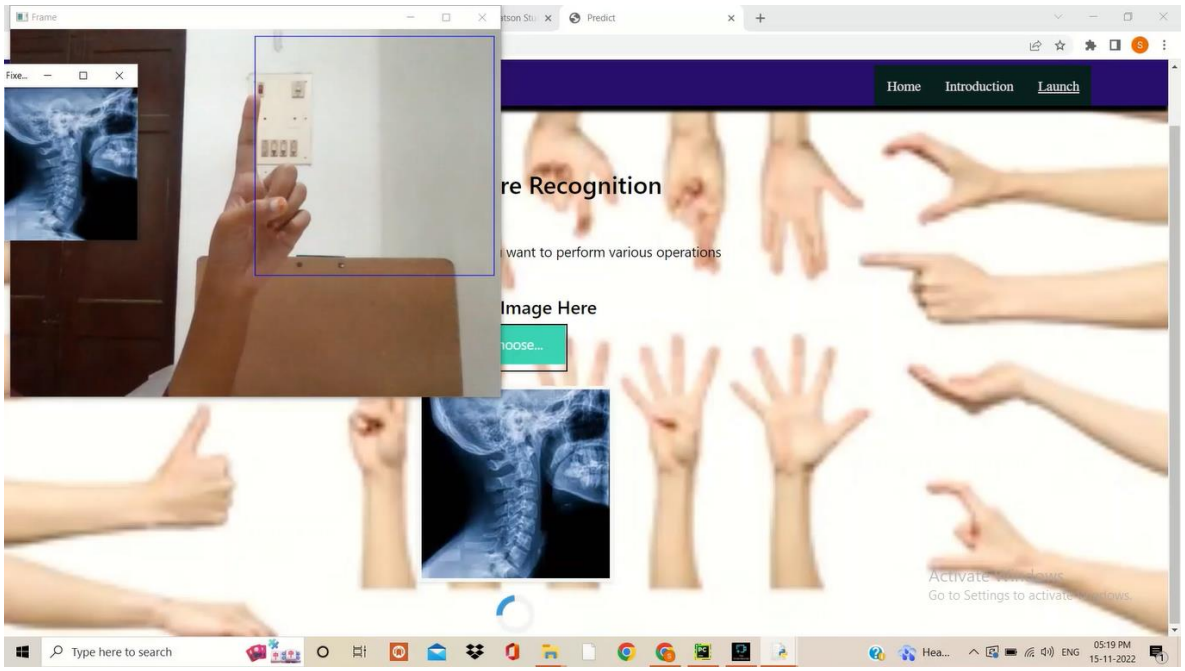
3:Predict Page:

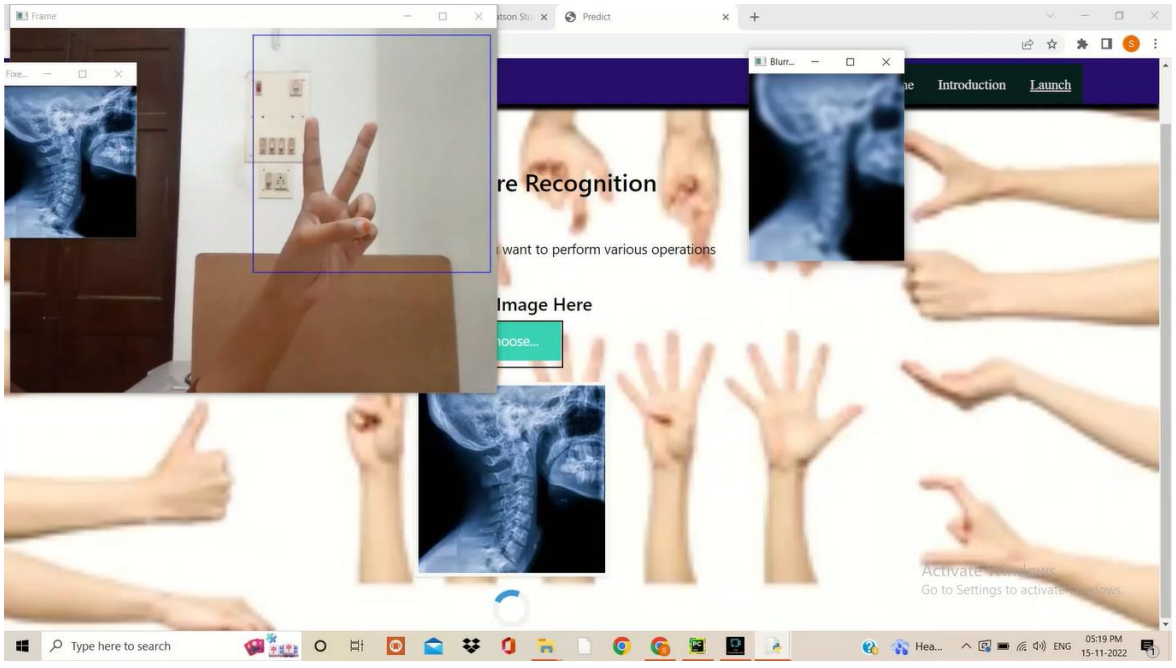


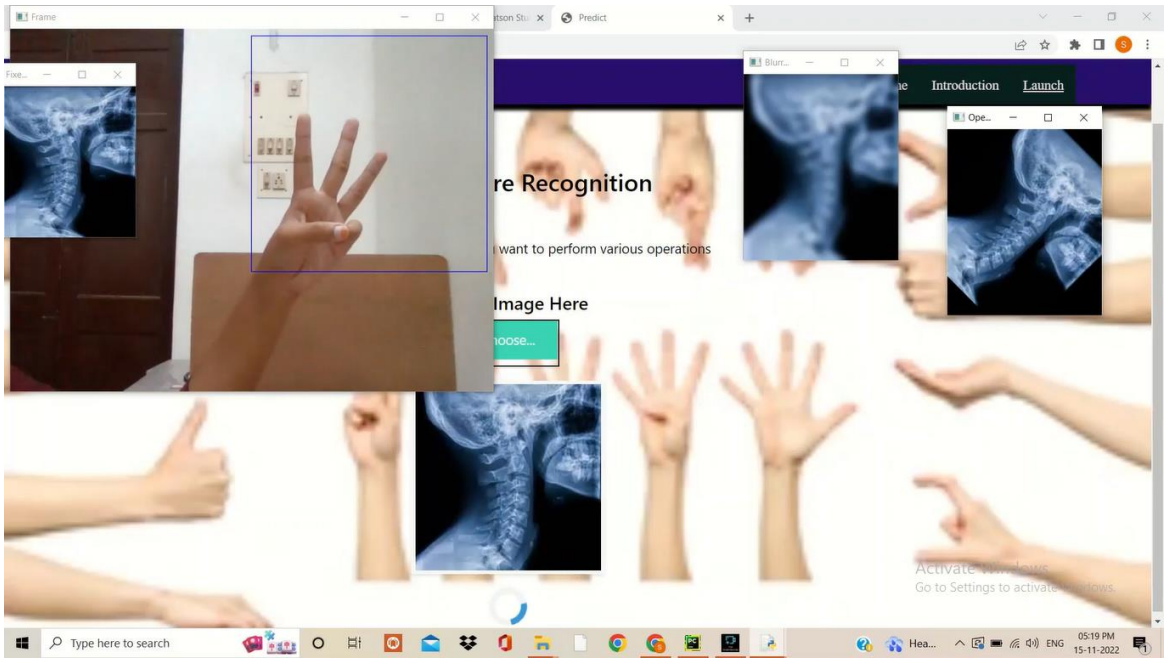
3.1 :Image Uploaded:

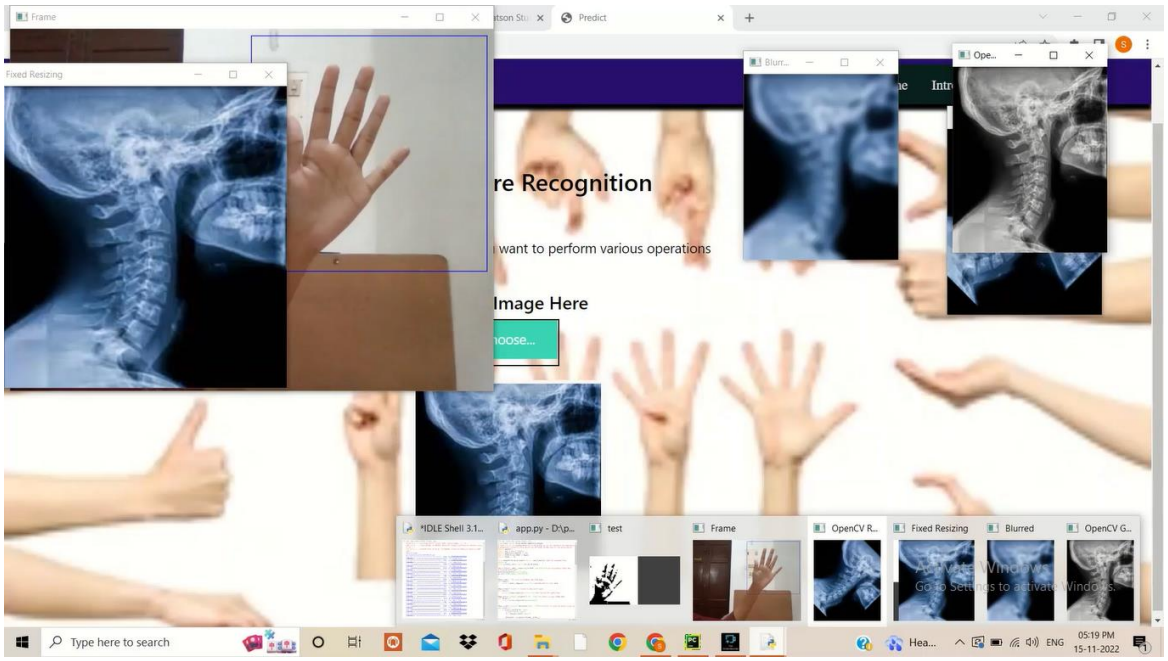


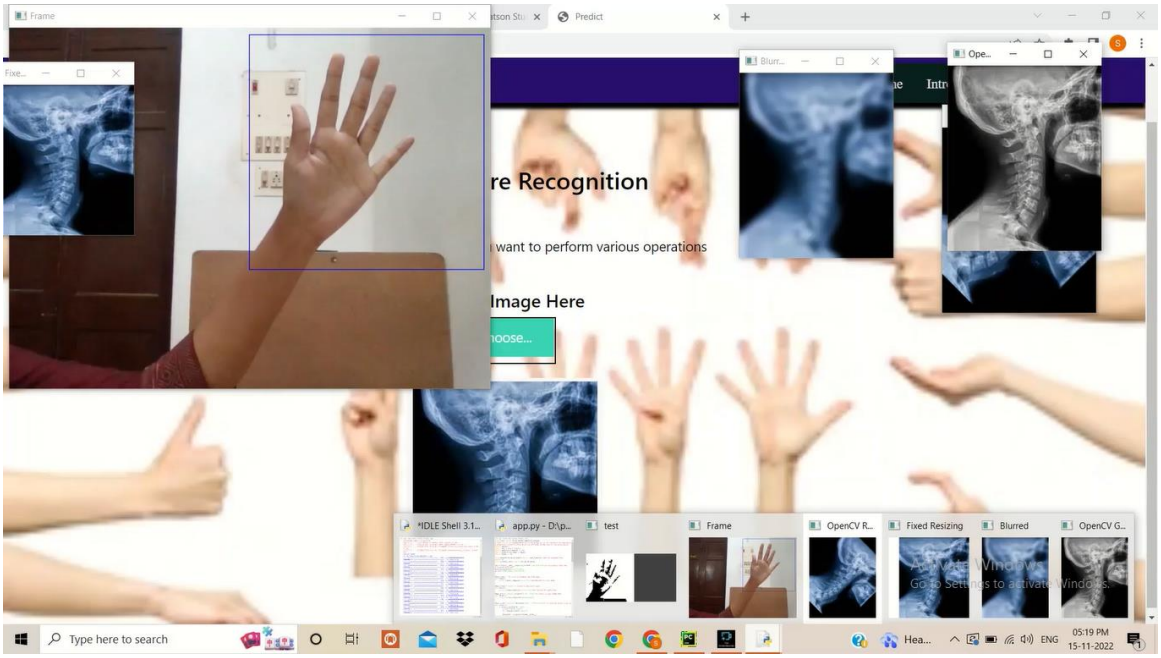
3.1 :Predict:


Frame
Predict
Home
Introduction
Launch
re Recognition
want to perform various operations
Image Here
Type here to search


Frame
Predict
Blurr...
Introduction
Launch
re Recognition
want to perform various operations
Image Here
Type here to search


Frame
Predict
Introduction
Launch
re Recognition
want to perform various operations
Image Here
Type here to search
05:19 PM
15-11-2022


Frame
Predict
Fixed Resizing
re Recognition
want to perform various operations
Image Here
*IDLE Shell 3.1...
app.py - D:\p...
test
Frame
OpenCV R...
Fixed Resizing
Blurred
OpenCV G...
Type here to search
05:19 PM
15-11-2022



4.Actions:

- ➢ 1-Fixed Resizing(200,200)
- ➢ 2- Blurred
- ➢ 3- OpenCV Rotation
- ➢ 4-Fixed Resizing(400,400)
- ➢ 5-OpenCV Grey Scale


Fixed Resizing


Ope...


Ope...

9.ADVANTAGES & DISADVANTAGES:

9.1.Advantages:

- **Ease of use**—the system allows the surgeon to use his/her hands, their natural work tool
- **Rapid reaction**—nonverbal instructions by hand gesture commands are intuitive and fast (In practice, the “Gestix” system can process images and track hands at a frame-rate of 150 Hz, thus, responding to the surgeon's gesture commands in real-time)
- **An unencumbered interface**—the proposed system does not require the surgeon to attach a microphone, use head-mounted (body-contact) sensing devices or to use foot pedals
- **Distance control**—the hand gestures can be performed up to 5 meters from the camera and still be recognized accurately.
- Accuracy is high.
- Memory Saving. [due to cloud]
- An Unecumbered Interface.
- Distance Control.

  Translation independent.

9.2.Disadvantages

- This kind of input also raises issues that are not relevant with traditional input. On the user's side, these problems are to learn, to remember and to accurately execute gestures.
- The developer has to provide a system that correctly recognizes these gestures. Freeman et al. remarked that the observation of gestures does not suffice in order to learn them, as the observer is unable to differentiate relevant and irrelevant movements.
- The teaching of multi-touch and mid-air gestures is more difficult than that of single-touch gestures. In the case of the latter, the hand posture is irrelevant - users only need to follow a path correctly to perform a command.
- Image Framing is not accurate.
- Small number of dataset.
- Wrong prediction in low light.
- Accuracy is less.
- Dataset is not suitable for real time.

10.CONCLUSION :

- In this project , we proposed an idea for feasible communication between hearing impaired and normasl person with the help of deep learning and machine learning approach. This is ever the surrounding challenge to developa sign language system in data the collection remains invariant of the unconstraint environment. This project can be ectended to the real time data. Our method shows to have potential in tackling this problem employing a straight forward camera as a premade dataset has been used.as been used, in case sufficient substantial training information is given, which can be continuously done and included through the previously mentioned process.

11. FUTURE SCOPE :

- The use of doctor-computer interaction devices in the operation room (OR) requires new

modalities that support medical imaging manipulation while allowing doctors' hands to remain sterile, supporting their focus of attention, and providing fast response times.In
future everything become automated .

12. APPENDIX :

12.1 Source Code

1:home.html

```
<html>
<script>


</script>


<style>
.header {       position: relative;
                        top:0;
                        margin:0px;
                        z-index: 1;
                        left: 0px;
                        right: 0px;
                        position: fixed;
                        background-color:rgb(40, 15, 109) ;
                        color: rgb(100, 100, 100);
                        box-shadow: 0px 8px 4px rgb(40, 15, 109);
                        overflow: hidden;
                        padding-left:20px;
                        font-family: 'Times New Roman';
                        font-size: 2vw;
                        width: 100%;
                        height:8%;
                        text-align: center;
                }
                .topnav {
  overflow: hidden;
  background-color: #84d39e;
}

.topnav-right a {
  float: left;
  color: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}
.topnav-right a.active {
  background-color: #07201e;
  color: rgb(238, 226, 234);
}
```

```
.topnav-right a:hover {
  background-color: rgb(255, 255, 255);
  color: rgb(6, 27, 36);
}

.topnav-right {
  float: right;
  padding-right:100px;
}

body {

  background-color: rgb(232, 214, 252) ;
  background-repeat: no-repeat;
  background-size:cover;
  background-position: 0px 0px;
  }
  .button {
  background-color: #091425;
  border: none;
  color: rgb(181, 228, 236);
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 10px;
  border-radius: 16px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  display: inline-block;
  margin-bottom:18px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

button {
  background-color: #091425;
  color: rgb(181, 228, 236);
  padding: 14px 20px;
  margin-bottom:10px;
  border: none;
  cursor: pointer;
```

```
  width: 17%;
  border-radius:4px;
  font-family:Montserrat;
}

button:hover {
  opacity: 0.8;
}

.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: rgb(100, 100, 100);
}

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}

img.avatar {
  width: 30%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}

.home{
          margin:80px;

  width: 84%;
```

```
  height: 500px;
  padding-top:10px;
  padding-left: 30px;

}
.login{
        margin:80px;
        box-sizing: content-box;
  width: 84%;
  height: 420px;
  padding: 30px;
  border: 10px solid rgb(13, 53, 68);
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
 border: 10px solid rgb(13, 53, 68);
}

.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Caption text */
.text {
  color: #9ac0c0;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
```

```
}

.active {
  color: rgb(145, 216, 221);
}

/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}




@import url("https://fonts.googleapis.com/css?family=Luckiest+Guy");
/* BODY */
body {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  width: 100%;
  height: 100%;
  overflow: hidden;
  font-family: "Arial", cursive;
  -webkit-font-smoothing: antialiased;
}

::selection {
  background: transparent;
```

```
}
/* CLOUDS */
body:before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  width: 0;
  height: 0;
  margin: auto;
  border-radius: 100%;
  background: transparent;
  display: block;
  box-shadow: 0 0 150px 100px rgba(255, 255, 255, 0.6),
    200px 0 200px 150px rgba(255, 255, 255, 0.6),
    -250px 0 300px 150px rgba(255, 255, 255, 0.6),
    550px 0 300px 200px rgba(255, 255, 255, 0.6),
    -550px 0 300px 200px rgba(255, 255, 255, 0.6);
}
/* JUMP */
h1 {
  cursor: default;
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  width: 100%;
  height: 100px;
  margin: 70px;
  display: block;
  text-align: center;
}

h1 span {
  position: relative;
  top: 5px;
  display: inline-block;
  font-size: 25px;
  color: #061a1f;
  text-shadow: 0 1px 0 rgb(207, 117, 245), 0 2px 0 rgb(207, 117, 245), 0 3px 0 rgb(207,
117, 245), 0 4px 0 rgb(207, 117, 245),
    0 5px 0 rgb(207, 117, 245), 0 6px 0 transparent, 0 7px 0 transparent, 0 8px 0
transparent,
    0 9px 0 transparent, 0 10px 10px rgba(58, 159, 167, 0.4);
}

h1 span:nth-child(2) {
```

```
  -webkit-animation-delay: 0.1s;
}

h1 span:nth-child(3) {
  -webkit-animation-delay: 0.2s;
}

h1 span:nth-child(4) {
  -webkit-animation-delay: 0.3s;
}

h1 span:nth-child(5) {
  -webkit-animation-delay: 0.4s;
}

h1 span:nth-child(6) {
  -webkit-animation-delay: 0.5s;
}

h1 span:nth-child(7) {
  -webkit-animation-delay: 0.6s;
}

h1 span:nth-child(8) {
  -webkit-animation-delay: 0.2s;
}

h1 span:nth-child(9) {
  -webkit-animation-delay: 0.3s;
}

h1 span:nth-child(10) {
  -webkit-animation-delay: 0.4s;
}

h1 span:nth-child(11) {
  -webkit-animation-delay: 0.5s;
}

h1 span:nth-child(12) {
  -webkit-animation-delay: 0.6s;
}

h1 span:nth-child(13) {
  -webkit-animation-delay: 0.7s;
}

h1 span:nth-child(14) {
  -webkit-animation-delay: 0.8s;
```

```
}

/* ANIMATION */
@-webkit-keyframes bounce {
  100% {
    top: -20px;
    text-shadow: 0 1px 0 #ccc, 0 2px 0 #ccc, 0 3px 0 #ccc, 0 4px 0 #ccc,
      0 5px 0 #ccc, 0 6px 0 #ccc, 0 7px 0 #ccc, 0 8px 0 #ccc, 0 9px 0 #ccc,
      0 50px 25px rgba(0, 0, 0, 0.2);
  }
}

</style>

<body>
<h1 style="color: rgb(193, 207, 207);">
  <table style="width:100%">
    <tr>

      <th> <iframe width="780" height="440"
src="https://www.youtube.com/embed/nD621G8u6oc?start=3&loop=1&autoplay=1&m
ute=1&controls=0">
      </iframe><br></th>
      <th></th>
    </tr>
  </table>
    <br>
    <span>HAND   GESTURE   RECOGNITION</span>
  </br>
    <span>OF RADIOLOGY   IMAGES</span>

  </br>
    <span>THROUGH STERILE   BROWSING</span>


</h1>
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:#c4dfd7; padding-
top:1%;padding-left:5%;"><b>HAND GESTURE RECOGNITION</b></div>
  <div class="topnav-right"style="padding-top:0.5%;color:white">
    <a  class="active" href="{{ url_for('home')}}"><u>Home</u></a>
    <a  class="active" href="{{ url_for('intro')}}">Introduction</a>
    <a  class="active" href="{{ url_for('image1')}}">Launch</a>
  </div>
</div>

</body>

</html>
```

2:intro.html

```
<html>
<script>


</script>


<style>
.header {       position: relative;
                        top:0;
                        margin:0px;
                        z-index: 1;
                        left: 0px;
                        right: 0px;
                        position: fixed;
                        background-color: rgb(40, 15, 109) ;
                        color: white;
                        box-shadow: 0px 8px 4px black;
                        overflow: hidden;
                        padding-left:20px;
                        font-family: 'Josefin Sans';
                        font-size: 2vw;
                        width: 100%;
                        height:8%;
                        text-align: center;
                }
                .topnav {
  overflow: hidden;
  background-color: #FCAD98;
}

.topnav-right a {
  float: left;
  color: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

.topnav-right a.active {
  background-color: #07201e;
  color: rgb(238, 226, 234);
}

.topnav-right a:hover {
  background-color: rgb(255, 255, 255);
  color: rgb(6, 27, 36);
```

```
}

.topnav-right {
  float: right;
  padding-right:100px;
}

body {

  background-color: ;
  background-repeat: no-repeat;
  background-size: contain;
  background-image: url("https://image.shutterstock.com/image-photo/multiple-womans-
hand-gestures-isolated-260nw-732833362.jpg");

    background-size: cover;
  background-position: 0px 0px;

  }
  .button {
  background-color: #091413;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 12px;
  border-radius: 16px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  display: inline-block;
  margin-bottom:18px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

button {
  background-color: #091425;
  color: white;
  padding: 14px 20px;
  margin-bottom:10px;
  border: none;
```

```
  cursor: pointer;
  width: 17%;
  border-radius:4px;
  font-family:Montserrat;
}

button:hover {
  opacity: 0.8;
}

.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}

img.avatar {
  width: 30%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}

.home{
          margin:80px;

  width: 84%;
```

```
  height: 500px;
  padding-top:10px;
  padding-left: 30px;

}
.login{
        margin:80px;
        box-sizing: content-box;
  width: 84%;
  height: 420px;
  padding: 30px;
  border: 10px solid blue;
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
 border: 10px solid blue;
}

.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}
```

```
.active {
  background-color: #FCAD98;
}

/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}



@import url("https://fonts.googleapis.com/css?family=Montserrat&display=swap");

* {
  padding: 0;
  margin: 0;
}

body {
  height: 100vh;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}

h1 {
  font-family: "Montserrat Medium";
  max-width: 90ch;
  text-align: center;
  transform: scale(0.94);
```

```
  animation: scale 3s forwards cubic-bezier(0.5, 1, 0.89, 1);
}
@keyframes scale {
  100% {
    transform: scale(1);
  }
}

span {
  display: inline-block;
  opacity: 0;
  filter: blur(4px);
}

span:nth-child(1) {
  animation: fade-in 1s 0.1s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(2) {
  animation: fade-in 0.8s 0.2s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(3) {
  animation: fade-in 0.8s 0.3s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(4) {
  animation: fade-in 0.8s 0.4s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(5) {
  animation: fade-in 0.8s 0.5s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(6) {
  animation: fade-in 0.8s 0.6s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(7) {
  animation: fade-in 0.8s 0.7s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(8) {
  animation: fade-in 0.8s 0.8s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(9) {
  animation: fade-in 0.8s 0.9s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
```

```
span:nth-child(10) {
  animation: fade-in 0.8s 1s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(11) {
  animation: fade-in 0.8s 1.1s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(12) {
  animation: fade-in 0.8s 1.2s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(13) {
  animation: fade-in 0.8s 1.3s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(14) {
  animation: fade-in 0.8s 1.4s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(15) {
  animation: fade-in 0.8s 1.5s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(16) {
  animation: fade-in 0.8s 1.6s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(17) {
  animation: fade-in 0.8s 1.7s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(18) {
  animation: fade-in 0.8s 1.8s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
span:nth-child(19) {
  animation: fade-in 0.8s 1.9s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
span:nth-child(20) {
  animation: fade-in 0.8s 2.0s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
span:nth-child(21) {
  animation: fade-in 0.8s 2.1s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
span:nth-child(22) {
  animation: fade-in 0.8s 2.2s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
span:nth-child(23) {
  animation: fade-in 0.8s 2.3s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(24) {
```

```
  animation: fade-in 0.8s 2.4s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(25) {
  animation: fade-in 0.8s 2.5s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(26) {
  animation: fade-in 0.8s 2.6s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(27) {
  animation: fade-in 0.8s 2.7s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(28) {
  animation: fade-in 0.8s 2.8s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
@keyframes fade-in {
  100% {
    opacity: 1;
    filter: blur(0);
  }
}

</style>

<body>
  <h1>INTRODUCTION</h1>
<h1>
  <span> The use of doctor-computer </span> <span> interaction devices </span >
<span>in the operation room </span>
    <span> requires new modalities </span> <span> that support medical imaging
</span> <span> manipulation while </span> <span> allowing doctors' hands </span>
<span>to remain sterile, </span> <span> supporting their </span > <span> focus of
attention, </span > <span> and providing fast response times. </span>
    <span> A gesture-based tool </span> <span>for sterile browsing  </span> <span> of
radiology images is a vision-based  </span> <span>hand gesture capture and   </span >
<span>recognition system that interprets  </span>
<span>in real-time the user's </span> <span>gestures for navigation  </span> <span>and
manipulation of images </span > <span>in an EMR database.</span> <span>The image
of the gesture </span> <span> captured in the video frame </span> <span>is compared
with </span > <span>the Pre-trained model </span>
<span> and the gesture is identified.</span> <span> If the gesture predictes is 1 </span>
<span>then images is blurred;2, </span > <span>image is resized;3,image is rotated
etc.</span>

</h1>
<!--Brian Tracy-->

<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:#c1e2d9; padding-
top:1%;padding-left:5%;">Hand Gesture System</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a class="active" href="{{ url_for('intro')}}"><u>Introduction</u></a>
    <a class="active" href="{{ url_for('image1')}}">Launch</a>
```

```
      </div>
    </div>
    </body>

    </html>
```

3:launch.html

```
<html lang="en">

  <head>
  <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=0.6">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <meta charset="UTF-8">
    <title>Predict</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
     rel="stylesheet">

    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
  <style>
  .bar
  {
  margin: 0px;
  padding:20px;
  background-color:black;
  opacity:0.6;
  color:black;
  font-family:'Roboto',sans-serif;
  font-style: italic;
  border-radius:20px;
  font-size:25px;
  }
  a
  {
  color:black;
  float:right;
  text-decoration:none;
  font-style:normal;
  padding-right:20px;
  }
  a:hover{
  background-color:black;
  color:black;
  font-size:30px;
  padding-left:10px;
```

```
}

div1{
  text-align: center;
  width: 650spx;
  height: 800px;
  padding: 190px;
  margin: 10px;
  position: absolute;
}



body
{
    background-image: url("https://image.shutterstock.com/image-photo/multiple-womans-
     hand-gestures-isolated-260nw-732833362.jpg");
    background-size: cover;
}

.header {position: relative;
      top:0;
      margin:0px;
      z-index: 1;
      left: 0px;
      right: 0px;
      position: fixed;
      background-color: rgb(40, 15, 109) ;
      color: black;
      box-shadow: 0px 8px 4px black;
      overflow: hidden;
      padding-left:20px;
      font-family: 'Josefin Sans';
      font-size: 2vw;
      width: 100%;
      height:8%;
      text-align: center;
    }
    .topnav {
  overflow: hidden;
  background-color: #051945;
}

.topnav-right a {
  float: left;
  color: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
```

```
}

.topnav-right a.active {
  background-color: #07201e;
  color: rgb(238, 226, 234);
}

.topnav-right a:hover {
  background-color: rgb(255, 255, 255);
  color: rgb(6, 27, 36);
}



.topnav-right {
  float: right;
  padding-right:100px;
}


.button {
background-color: #091425;
border: none;
color: black;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 12px;
border-radius: 16px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 2px solid black; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  display: inline-block;
  margin-bottom:18px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

button {
  background-color: #091425;
  color: black;
  padding: 14px 20px;
  margin-bottom:10px;
```

```
  border: none;
  cursor: pointer;
  width: 17%;
  border-radius:4px;
  font-family:Montserrat;
}

button:hover {
  opacity: 0.8;
}

.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}

img.avatar {
  width: 30%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}

.home{
  margin:80px;
```

```
  width: 84%;
  height: 500px;
  padding-top:10px;
  padding-left: 30px;

}
.login{
  margin:80px;
  box-sizing: content-box;
  width: 84%;
  height: 420px;
  padding: 30px;
  border: 10px solid rgb(12, 91, 94);
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
 border: 10px solid rgb(12, 91, 94);
}

.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
```

```
}

.active {
  background-color: #267481;
}

/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}

</style>
</head>

<body>
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:#c1e2d9; padding-
    top:1%;padding-left:5%;">Hand Gesture System</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a class="active" href="{{ url_for('intro')}}">Introduction</a>
    <a class="active" href="{{ url_for('image1')}}"><u>Launch</u></a>
  </div>
</div>
<br>

  <div1 style=""><h1><font color="Black" size="6" font-family="Roboto">Hand Gesture
    Recognition</h1><br>
  <p><i><font color="Black" size="4" fonr-family="sans-serif"></i>Choose an image for
    which you want to perform various operations</p>
  <br>
```

```
        <div>
          <h4>Upload Image Here</h4>
      <form action = "http://localhost:5000/" id="upload-file" method="post"
     enctype="multipart/form-data">
        <label for="imageUpload" class="upload-label">
          Choose an image
        </label>
        <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg,.pdf">
      </form>
           <center>
      <div class="image-section" style="display:none;">
        <div class="img-preview">
          <div id="imagePreview">
          </div>
        </div>
        <div>
          <button type="button" class="btn btn-info btn-lg " id="btn-
     predict">Predict!</button>
        </div>
      </div>
      <div class="loader" style="display:none;"></div>
           </center>
    </div>
    </div1>

        <footer>
    <script src="{{ url_for('static', filename='js/main.js') }}"
      type="text/javascript"></script>
</footer>

</html>
```

12.2 GitHub & Project Demo Link

PROJECT GITHUB:

https://github.com/IBM-EPBL/IBM-Project-7007-1658845002

PROJECT DEMO LINK:

https://drive.google.com/file/d/14hVUvkyUSb_oD9Xl98C4W8AKk5aY8yOs/view?usp=share_link