A project report on

Gesture Based Tool for Sterile Browsing of Radiology Images

Powered by IBM India

Project Id: PNT2022TMID07081

Submitted by

Maharaja S (Team Lead) 130719104048

Karthikeyan L 130719104040

Akhil P Aji 130719104001

Kathirvelan CR 130719104042

Department of Computer Science Engineering Jerusalem College of Engineering

Velachery main road, Narayanapuram, Chennai - 600100 Industrial Mentor: Pradeepthi

1. INTRODUCTION

1.1 Overview

In this project we use gestures to browse radiology images. Gestures refer to non-verbal form of communication.

A major challenge involved in this process is to provide doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work. Keyboards and pointing devices, such as a mouse, are today's common method of human—computer interaction.

However, the use of computer keyboards and mouse by doctors and nurses in intensive care units (ICUs) is a common method for spreading infections.

Humans can recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development.

In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others. In this project Gesture based Desktop automation, First the model is trained pre trained on the images of different hand gestures, such as a showing numbers with fingers as 1,2,3,4. This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the gesture predicts is 0 - then images is converted into rectangle, 1 - image is Resized , 2 - image is rotated, 3 - image is blurred.

1.2 PURPOSE

Computer information technology is increasingly penetrating into the hospital domain. A major challenge involved in this process is to provide doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work. Keyboards and pointing devices, such as a mouse, are today's principal method of human—computer interaction. However, the use of computer keyboards and mice by doctors and nurses in intensive care units (ICUs) is a common method for spreading infections.we suggest the use of hand gestures as an alternative to existing interface techniques, offering the major advantage of sterility In this work we refer to gestures as a basic form of non-verbal communication made with the hands. Psychological studies showed that young children use gestures to communicate before they learn to talk. Manipulation, as a form of gesticulation, is often used when people speak to each other about some object. Naturalness of expression, non-encumbered interaction, intuitiveness and high sterility are all good reasons to replace the current interface technology (e.g., keyboard, mouse, and joystick) with more natural interfaces.In simply It is used to browse through the images obtained using radiology using hand gestures rather than using mouse,keyboard,etc thereby maintaining sterility.

2. LITERATURE SURVEY

2.1 Existing problem

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 Keyboards and pointing devices, suchas a mouse, are today's principal method of human—computer interaction. However, the use of computer key-boards and mice by doctors and nurses in intensive care units (ICUs) is a common method for spreading infections. In this paper, we suggest the use of hand gestures as an alternative to existing interface techniques, offering themajor advantage of sterility. Even though voice control alsoprovides sterility, the noise level in the operating room (OR)deems it problematic.

2.2 Project Objecves

- ➢ Know fundamental concepts and techniques of Convolutional Neural Network (CNN).
- ➢ Gain a broad understanding of image data.
- ➢ Know how to pre-process/clean the data using different data preprocessing techniques.
- ➢ Know how to build a web application using Flask framework.

2.3 References

1. Qing Chen Nicolas, D. Georganas, and Emil M. Petriu "Hand Gesture Recognition Using Haar-Like Features And A Stochastic Context-Free Grammar" IEEE ,Vol. 57, No. 8, August 2008

2. Anupam Agrawal, Rohit Raj and Shubha Porwal "Vision-based Multimodal HumanComputer Interaction using Hand and Head Gestures" IEEE Conference on Information and Communication Technologies ICT 2013

3. Kenji Oka and Yoichi Sato "Real-Time Fingertip Tracking and Gesture Recognition" IEEE proceeding on Computer Graphics and Applications Nov/Dec 2002

4. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International Conference on Machine Learning, 2015, pp. 448–456.

5. Juan Wachs, Helman Stern, Yael Edan, Michael Gillam, Jon Handler, Craig Feied, Mark Smith

6. Nishikawa A, Hosoi T, Koara K, Negoro D, Hikita A, Asano S, Kakutani H, Miyazaki F, Sekimoto M, Yasui M, Miyake Y, Takiguchi S, Monden M. "Face MOUSe: A Novel Human Machine Interface for Controlling the Position of a Laparoscope," IEEE Trans. on Robotics and Automation 2003;19(5):825- 841.

7. Smith KR, Frank KJ, Bucholz RD. "The NeuroStation- a highly accurate, minimally invasive solution to frameless stereotatic neurosurgery," Comput Med Imaging Graph 1994;18:247-256.

8. Graetzel C, Fong TW, Grange S, Baur C. "A non-contact mouse for surgeon-computer interaction," Technol Health Care 2004;12(3):245-257.

3. THEORITICAL ANALYSIS

Date	20 Oct 2022
Team ID	PNT2022TMID07081
Project Name	Project - **A Gesture-based Tool for Sterile Browsing of Radiology Images**

Solution Architecture:

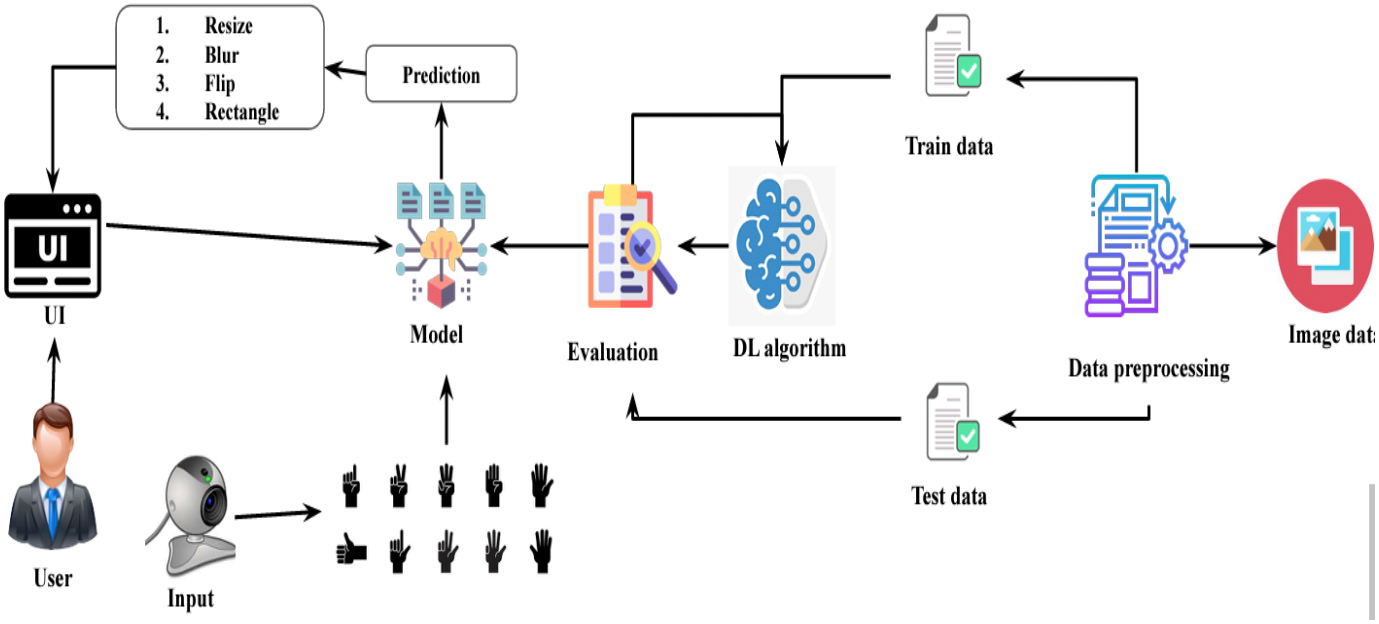


Fig 3. Architecture of Gesture Based Tool for Sterile Browsing of Radiology Images

4. EXPERIMENTAL INVESTIGATIONS

We found that many hospitals rely on mouse and keyboard to browse the images that are obtained during different surgeries, scans, etc. This can contaminate the environment with various infections thus compromising the sterility.

Various technologies have been developed to overcome this issue and one such technology was called ‘Gestix’.

This hand gesture system for MRI manipulation in an EMR image database called “*Gestix*” was tested during a brain biopsy surgery. This system is a real-time hand-tracking recognition technique based on color and motion fusion. In an in vivo experiment, this type of interface prevented the surgeon's focus shift and change of location while achieving rapid intuitive interaction with an EMR image database. In addition to allowing sterile interaction with EMRs, the “*Gestix*” hand gesture interface provides:

1. ease of use—the system allows the surgeon to use his/her hands, their natural work tool

2. rapid reaction—nonverbal instructions by hand gesture commands are intuitive and fast

3. 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

4. distance control—the hand gestures can be performed up to 5 meters from the camera and still be recognized accurately.

5. PROJECT FLOW

➢ User interacts with the UI (User Interface) to upload the image as input.

➢ Depending on the different gesture inputs different operations are applied to the input image.

➢ Once model analyses the gesture, the prediction with operation applied on image is showcased on the UI.

To accomplish this, we have to complete all the activities and tasks listed below:

➢ Data Collection.

- Collect the dataset or Create the dataset

➢ Data Pre processing

- Import the ImageDataGenerator library
- Configure ImageDataGenerator class
- Apply ImageDataGenerator functionality to Trainset and Testset

➢ Model Building

- Import the model building Libraries
- Initializing the model
- Adding Input Layer
- Adding Hidden Layer
- Adding Output Layer

- Configure the Learning Process
- Training and testing the model
- Save the Model

➢ Application Building

- Create an HTML file
- Build Python Code

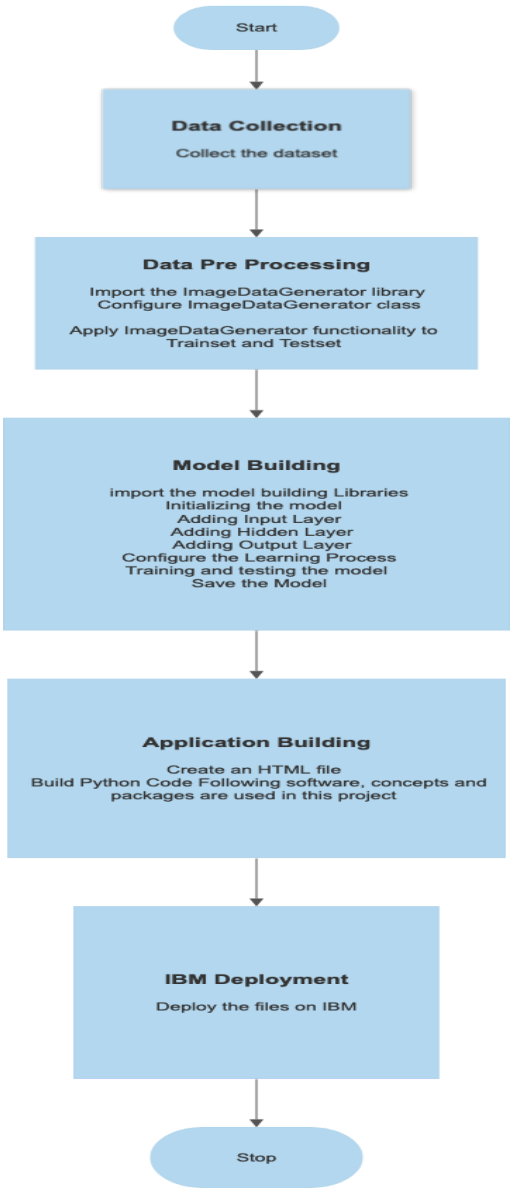
Following software, concepts and packages are used in this project

➢ Anaconda navigator

➢ Python packages:

- open anaconda prompt as administrator
- Type “pip install TensorFlow” (make sure you are working on python 64 bit)
- Type “pip install opencv-python”
- Type “pip install flask

6.FLOW CHART


Start
Data Collection
Collect the dataset
Data Pre Processing
Import the ImageDataGenerator library
Configure ImageDataGenerator class
Apply ImageDataGenerator functionality to
Trainset and Testset
Model Building
import the model building Libraries
Initializing the model
Adding Input Layer
Adding Hidden Layer
Adding Output Layer
Configure the Learning Process
Training and testing the model
Save the Model
Application Building
Create an HTML file
Build Python Code Following software, concepts and
packages are used in this project
IBM Deployment
Deploy the files on IBM
Stop

- ”

7.Deep Learning Concepts :

CNN:

A convoluonal neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery.

Opencv:

It is an Open Source Computer Vision Library which are mainly used for image processing, video capture and analysis including features like face detecon and object detecon.

Flask:

Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applicaons

8.Project Structure :

- ❖ Dataset folder contains the training and testing images for training our model.

- ❖ We are building a Flask Application which needs HTML pages stored in the templates folder and a python script app.py for server side scripting

- ❖ We need the model which is saved and the saved model in this content is gesture.h5

- ❖ The static folder will contain js and css files

- ❖ Whenever we upload a image to predict, that images is saved in uploads folder.

9.APPLICATION BUILDING

BUILD HTML PAGES

- ★ We use HTML to create the front end part of the web page.
- ★ Here, we created 3 html pages- home.html, intro.html and index.html
- ★ home.html displays home page.
- ★ Intro.html displays introduction about the hand gesture recognition
- ★ index6.html accepts input from the user and predicts the values.
- ★ We also use CSS-style.css to enhance our functionality and view of HTML page

HTML Programs

home.html

```
<!DOCTYPE html>
<style>
body{
 background-color: ;
 background-repeat: no-repeat;
 background-size:cover;
 background-image:
url("https://cdn.dribbble.com/users/662638/screenshots/4803914/media/23c3ff6eda9b508b7c079b081a6a9985.gif");
 background-size: cover;
 background-position: 0px 0px;
}
</style>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
   </head>
<body>
```

```
 <nav>
  <div class="menu">
   <div class="logo">
    <a href="#">HAND GESTURE RECOGNITION</a>
   </div>
   <ul>
    <li><a href=home.html>Home</a></li>
    <li><a href=intro.html>Intro</a></li>
    <li><a href=index.html>Launch</a></li>
   </ul>
  </div>
 </nav>
 <h1 style="color: rgb(193, 207, 207);">
 <div class="center">
  <div class="title">A Gesture-based Tool for Sterile Browsing of Radiology Images</div>
  <div class="sub_title">DONE BY : Karthikeyan.L , Akhil.p.aji , Maharaja , Kathirvelan.
 </div>
</body>
</html>
```

intro.html

```
<!DOCTYPE html>
<style>
body{
 background-color: ;
 background-repeat: no-repeat;
 background-size:cover;
 background-image:
url("https://images.squarespace-cdn.com/content/v1/5133ad10e4b03f8
a261f0ff2/1477933433805-XTQW6Z8W2AE229XY9N0S/intro+hand+
gif.gif?format=2500w");
 background-size: cover;
 background-position: 0px 0px;
}

</style>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
   </head>
<body>
```

```
<nav>
  <div class="menu">
    <div class="logo">
      <a href="#">HAND GESTURE RECOGNITION</a>
    </div>
    <ul>
       <li><a href=home.html>Home</a></li>
       <li><a href=intro.html>Intro</a></li>
      <li><a href=index.html>Launch</a></li>
    </ul>
  </div>
</nav>
<h1 style="color: rgb(193, 207, 207);">
  <div class="center">
    <div class="title">INTRODUCTION</div>
    <div class="sub_title">Hand Gesture recognition system provides us an innovative,natural, user friendly way of interaction with the computer which is more familiar to the human beings. In our project, the hand region > is extracted from the background by using Region of intrest. Then, we will be predicting the labels based on the CNN trained model weights of hand gestures using that predicted labels we apply if conditions to control some of the actions like reshaping , blur, flip of the given image
  </div>
</body>
```

```
</html>
```

index.html

```
<!DOCTYPE html>

<html>
<style>
        body{
         background-color: ;
         background-repeat: no-repeat;
         background-size:cover;
         background-image:
url("https://big.assets.huffingtonpost.com/dignity-health-02.gif");
         background-size: cover;
         background-position: 0px 0px;
        }
        @import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;
400;500;600;700&display=swap');

</style>
<head>
            <meta charset="UTF-8">
            <meta name="viewport" content="width=device-width,
initial-scale=1.0">
```

```
        <link rel="stylesheet" href="style.css">
 </head>



<body>
   <nav>
      <div class="menu">
        <div class="logo">
          <a href="#">HAND GESTURE RECOGNITION</a>
        </div>
        <ul>
          <li><a href=home.html>Home</a></li>
          <li><a href=intro.html>Intro</a></li>
         <li><a href=index.html>Launch</a></li>
        </ul>
      </div>
    </nav>
   <h1 style="color: rgb(9, 8, 8);">
      <div class="center">
        <div class="title">Upload Image Here</div>
        <div><font font-background="white" color="Black" size="6"
fonr-family="sans-serif"> Provide an image for which you want to
perform various operations
      </font>
```

```
    </div>


<form action="/action_page.php"
    enctype="multipart/form-data">


    <label for="myfile">Select a file:</label>


    <input type="file" id="myfile"
        name="myfile" multiple="multiple" />


    <br /><br />
```

```
        <input type="submit" />

    </form>

</body>




</html>
```

CSS Programs

style.css

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;
400;500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins',sans-serif;
}
```

```
::selection{
  color: #000;
  background: #fff;
}
nav{
  position: fixed;
  background: #1b1b1b;
  width: 100%;
  padding: 10px 0;
  z-index: 12;
}
nav .menu{
  max-width: 1250px;
  margin: auto;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 20px;
}
.menu .logo a{
  text-decoration: none;
  color: #fff;
  font-size: 35px;
  font-weight: 600;
}
```

```
.menu ul{
  display: inline-flex;
}
.menu ul li{
  list-style: none;
  margin-left: 7px;
}
.menu ul li:first-child{
  margin-left: 0px;
}
.menu ul li a{
  text-decoration: none;
  color: #fff;
  font-size: 18px;
  font-weight: 500;
  padding: 8px 15px;
  border-radius: 5px;
  transition: all 0.3s ease;
}
.menu ul li a:hover{
  background: #fff;
  color: black;
}
.img{
  background: url('img3.jpg')no-repeat;
```

```
  width: 100%;
  height: 100vh;
  background-size: cover;
  background-position: center;
  position: relative;
}
.img::before{
  content: '';
  position: absolute;
  height: 100%;
  width: 100%;
  background: rgba(0, 0, 0, 0.4);
}
.center{
  position: absolute;
  top: 52%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 100%;
  padding: 0 20px;
  text-align: center;
}
.center .title{
  color: #fff;
  font-size: 55px;
```

```
  font-weight: 600;
}
.center .sub_title{
  color: rgb(140, 143, 143);
  font-size: 25px;
  font-weight: 250;
  text-align: center;
}
```

BUILD PYTHON CODE

- ★ Build flask file 'app.py' which is a web framework written in python for server-side scripting.
- ★ 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.
- ★ Importing Libraries

PYTHON PROGRAM

app.py

```
from flask import Flask,render_template,request
import numpy as np
import os
import operator
import cv2
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
from werkzeug.utils import secure_filename
```

```
app = Flask(__name__,template_folder="templates")
model=load_model('gesture.h5')
print("Loaded model from disk")

@app.route("/")
def root():
        return render_template("home.html")

@app.route("/home")
def home():
        return render_template("home.html")

@app.route("/intro")
def intro():
        return render_template("intro.html")

@app.route("/index6")
def index6():
        return render_template("index6.html")


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

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

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

        basepath=os.path.dirname(__file__)

file_path=os.path.join(basepath,'uploads',secure_filename(f.filen
ame))
        f.save(file_path)
        print('img saved successfully')
        print(file_path)



    cap=cv2.VideoCapture(0)
    while True:
        _, frame=cap.read()
        frame=cv2.flip(frame,1)

        x1=int(0.5*frame.shape[1])
        y1=10
        x2=frame.shape[1]-10
        y2=int(0.5*frame.shape[1])
```

```
        cv2.rectangle(frame,(x1-1,y1-1),(x2+1,y2+1),(255,0,0)),1
        roi = frame[y1:y2,x1:x2]

        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)

        result = model.predict(test_image.reshape(1,64,64,1))
        print(result)
        prediction =
{'ZERO':result[0][0],'ONE':result[0][1],'TWO':result[0][2],'TH
REE':result[0][3],'FOUR':result[0][4],'FIVE':result[0][5]}

prediction=sorted(prediction.items(),key=operator.itemgetter(1),
reverse=True)

        cv2.putText(frame,prediction[0][0],(10,120),
cv2.FONT_HERSHEY_PLAIN,1,(0,255,255),1)
        cv2.imshow("frame",frame)

        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:
            break

    cap.release()
    cv2.destroyAllWindows()

    return render_template("home.html")

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

RUN THE APPLICATION

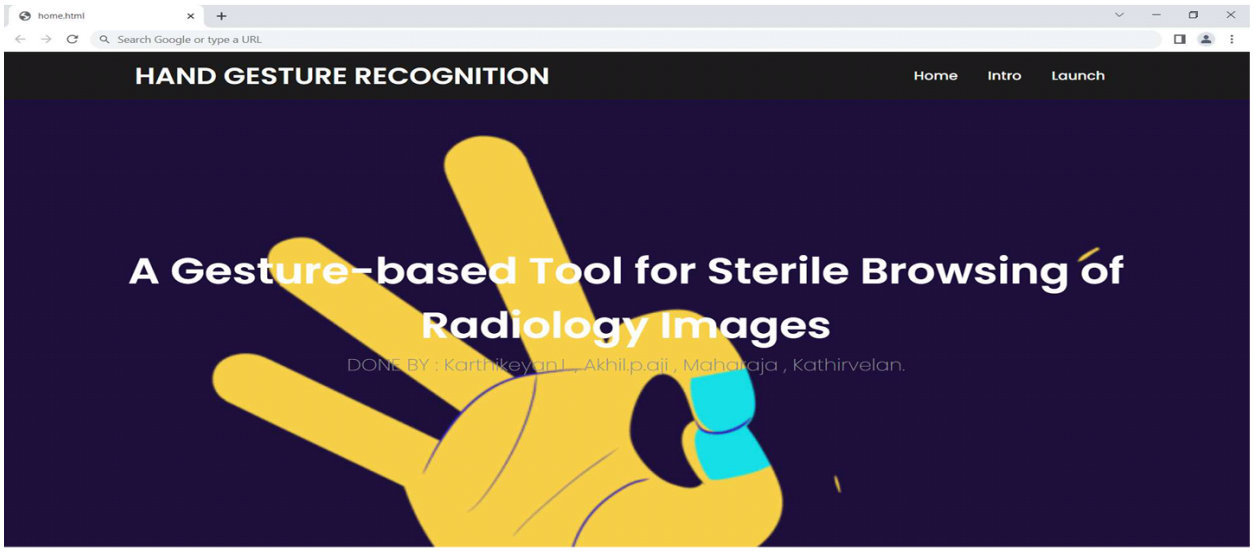
At last, we will run our flask applicaon

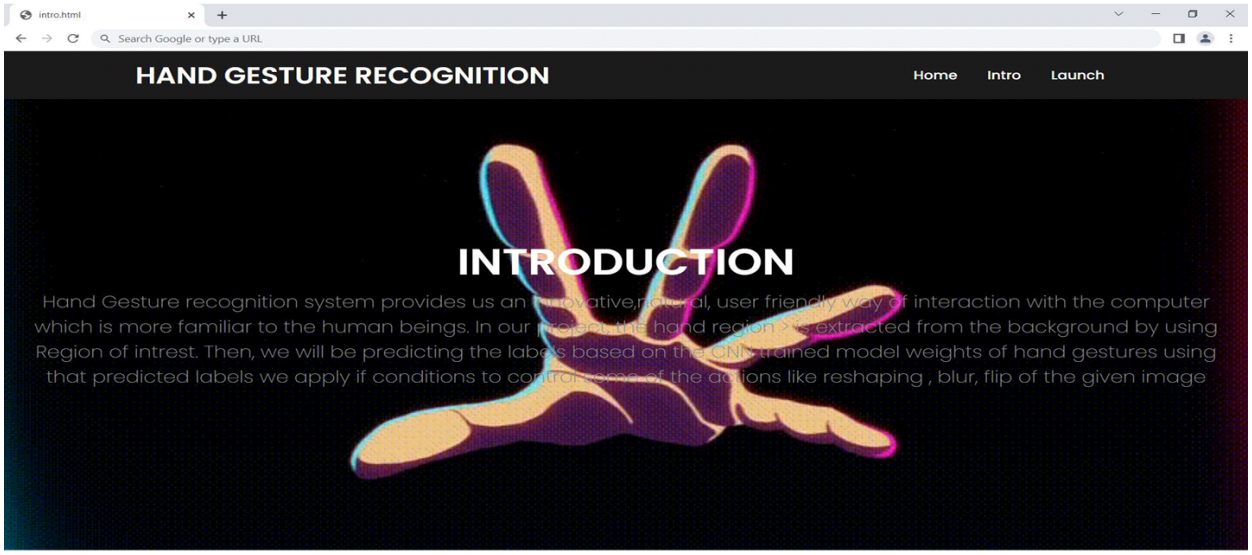
Run The app.py in local browser

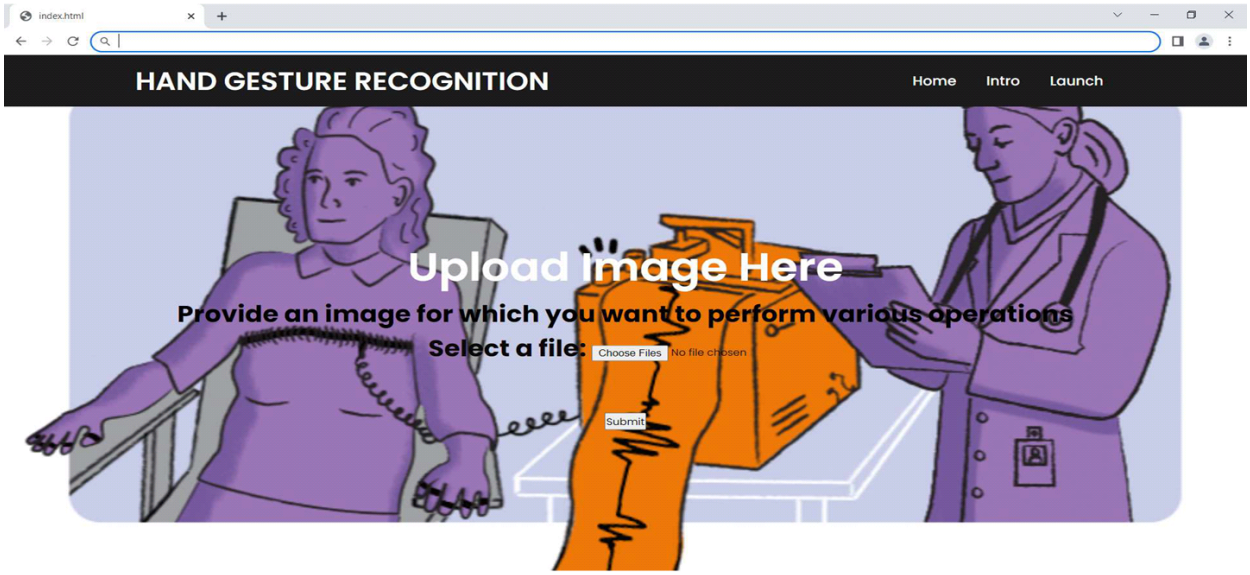
- **Open anaconda prompt from the start menu**
- **Navigate to the folder where your python script is.**
- **Now type "python app.py" command Navigate to the localhost where you can view your web page**
- **Navigate to the localhost where you can view your web page**
- **Then it will run on localhost:5000**
- **Navigate to the localhost (hp://127.0.0.1:5000/)where you can view your web page.**

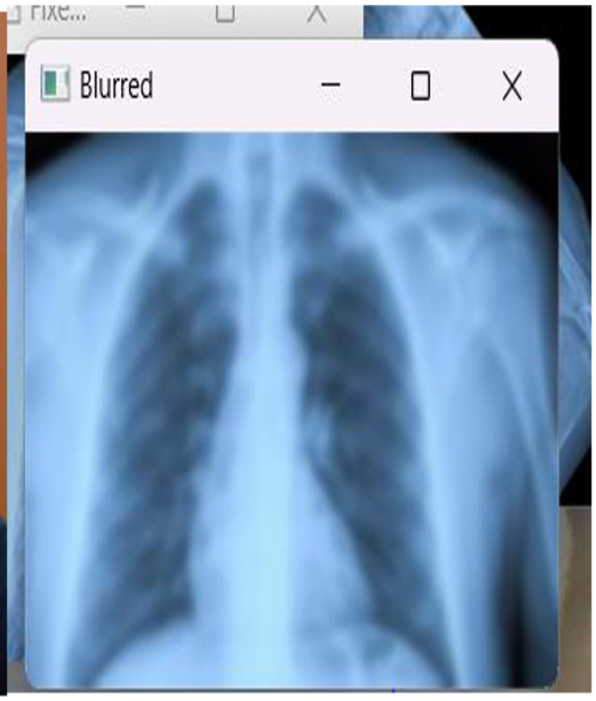
10. RESULT

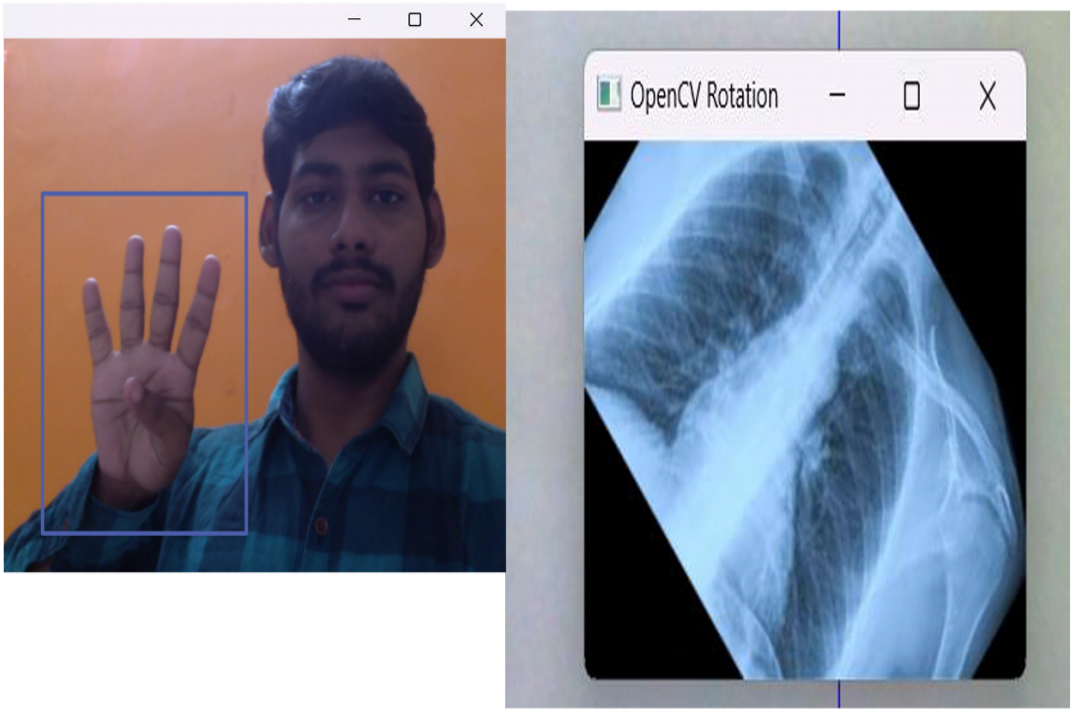
Final (Output) of the project along with screenshots.






index.html
HAND GESTURE RECOGNITION
Home
Intro
Launch
Upload Image Here
Provide an image for which you want to perform various operations
Select a file:
Choose Files
No file chosen
Submit


Blurred


OpenCV Rotation

11. ADVANTAGES & DISADVANTAGES

Advantages:

- Major advantage of this tool is that it helps to maintain the sterility of the environment.
- It is also easy to use and is quicker than the existing methods to browse images.
- It can also be performed even if the surgeon is a bit far away from the system, this helps to save time.
- The tool does not need the person using it to have an apparatus or any devices on them to use it. They can simply move their hands to browse through the images.

Disadvantages:

- The tool can be quite expensive as it requires cameras and other expensive devices to capture images and process it.
- Image rendering may happen frequently, this can frustrate the doctors during a procedure.
- There is a chance the hospital might lack the required technical infrastructure.
- Also, there is a chance that surgeons might show a constraints towards using this tool due to their lack of knowledge using these kinds of technological to tools

12. APPLICATIONS

- This hand based gesture tool developed can be mainly used in the medical industry to browse images without compromising the sterility.
- However it can also be used in different industries while presenting certain ideas, during meetings, and can be used by teachers while teteaching.

13. CONCLUSION

In this project we developed a tool which recognises hand gestures and enables doctors to browse through radiology images using these gestures. This enables doctors and surgeons to maintain the sterility as they would not have to touch any mouse or keyboard to go through the images.

This tool is also easy to use and is quicker than the regular method of using mouse/keyboard.

It can be used regardless of the users location since they don't have to be in contact with any device. It also does not require the user to have any device on them to use it.

Further this technology can be extended to other industries like it can be used by presenters, by teachers for
show images in the classroom, etc.

14. FUTURE SCOPE

- The tool can be made quicker by increasing the recognition speed.
- More number of gestures can be added thereby increasing this tool's functionality and useability for different purposes.
- Tracking of both hands can be added to increase the set of commands. Voice commands can also be added to further increase the functionality.

15. REFERENCE

1. Qing Chen Nicolas, D. Georganas, and Emil M. Petriu "Hand Gesture Recognition Using Haar-Like Features And A Stochastic Context-Free Grammar" IEEE ,Vol. 57, No. 8, August 2008
2. Anupam Agrawal, Rohit Raj and Shubha Porwal "Vision-based Multimodal HumanComputer Interaction using Hand and Head Gestures" IEEE Conference on Information and Communication Technologies ICT 2013
3. Kenji Oka and Yoichi Sato "Real-Time Fingertip Tracking and Gesture Recognition" IEEE proceeding on Computer Graphics and Applications Nov/Dec 2002
4. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International Conference on Machine Learning, 2015, pp. 448–456.
5. Juan Wachs, Helman Stern, Yael Edan, Michael Gillam, Jon Handler, Craig Feied, Mark Smith 6.Professor. Juan P. Wachs, 7.Professor. Benjamin Fritsch

16. Appendix

source code

Github Link - https://github.com/IBM-EPBL/IBM-Project-25712-1659971442