Gesture Based Tool for Sterile Browsing of Radiology Images

Nalaiyathiran

Project report

Team ID:PNT2022TMID04610

JANE SALINA MARIS J

MAHENDRAWARMAN K

HEMA S

KEERTHIRAJAN S

CONTENT

Gesture Based Tool for Sterile Browsing of Radiology Images

1 INTRODUCTION

1.1 Overview

In this project we use gestures to browse images obtained during radiology. Gestures refer to non-verbal form of communication made using hands. 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. 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 number 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 into (200,200), 2 - image is rotated by -45°, 3 - image is blurred, 4 - image is Resized into (400,400), 5 - image is converted into grayscale etc.

1.2 Purpose

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

A Gesture-based Tool for Sterile Browsing of Radiology Images - research paper by national library of medicine the hand gesture control system “Gestix” developed by the authors helped the doctor to remain in place during the entire operation, without any need to move to the main control wall since all the commands were performed using hand gestures. The sterile gesture interface consists of a Canon VC-C4 camera, whose pan/tilt/zoom can be initially set using an infrared (IR) remote. This camera is placed just over a large flat screen monitor. Additionally, an Intel Pentium IV, (600MHz, OS: Windows XP) with a Matrox Standard II video-capturing device is used. The “Gibson” image browser is a 3D visualization medical tool that enables examination of images, such as: MRIs, CT scans and X-rays. The images are arranged over a multiple layer 3D cylinder. The image of interest is found through rotating the cylinder in the four cardinal directions. To interface the gesture recognition routines with the “Gibson” system, information such as the centroid of the hand, its size, and orientation are used to enable screen operations in the “Gibson” graphical user interface

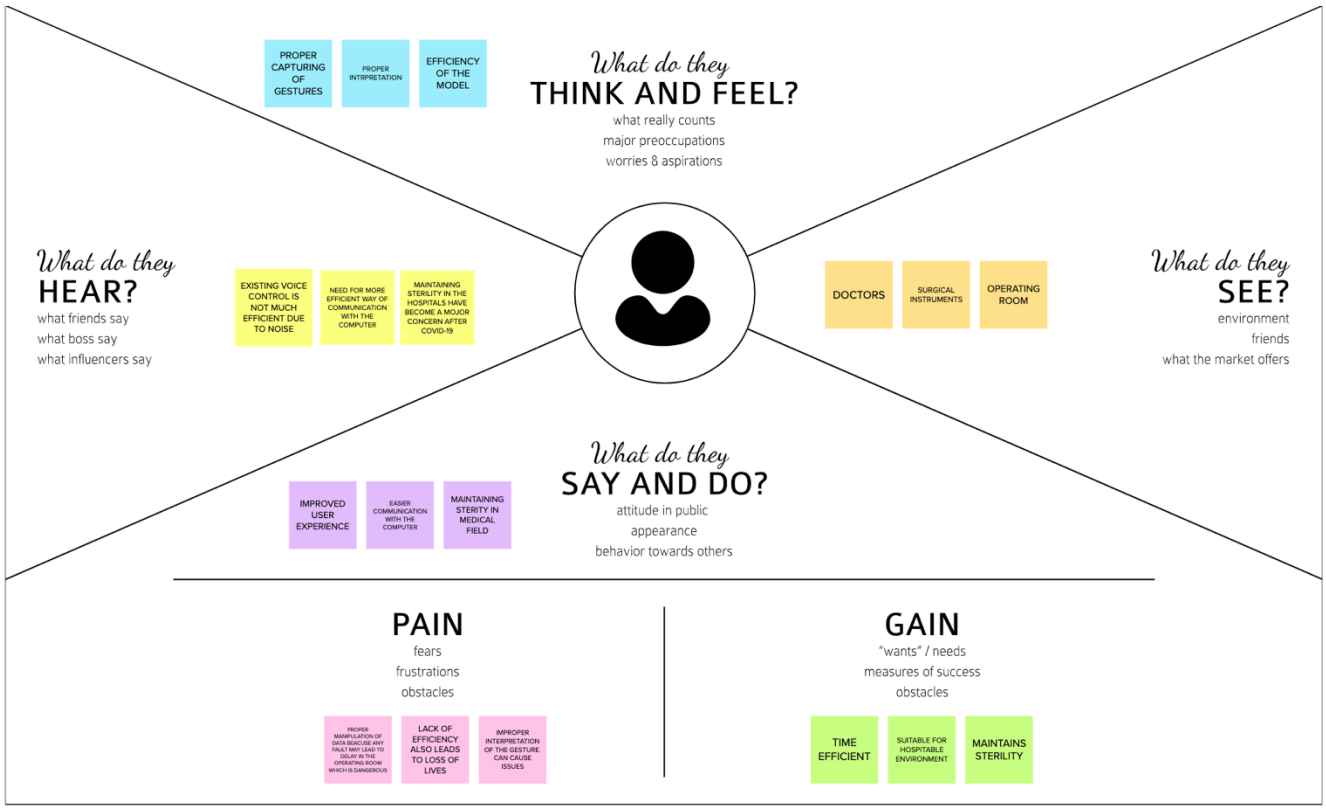
2.2 Problem Statement Definition

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. This paper presents“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. Navigation and other gestures are translated to commands based on their temporal trajectories, through video capture. “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. Data from two usability tests provide insights and implications regarding human-computer interaction based on nonverbal conversational modalities.

3 THEORITICAL ANALYSIS

3.1 Empathy Map Canvas

In this activity we prepared the empathy map canvas to capture the user Pains & Gains, Prepare list of problem statements.



3.1 Proposed Solution

In this we prepared the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.

S.No.	Parameter	Description
1.	Problem Statement	At present Doctors are interacting with system via hands which will be infection via germs during the operation, thus we are going to the contactless navigation of radiology images for treatment using ML model to identify the gestures.
2.	Idea / Solution description	To avoid the contact gesture- b a s e d communication is implemented using CNN and Cameras which detects the gestures, W h i c h will be absolutely sterile.
3.	Novelty / Uniqueness	We are going to use the CNN for recognizing the gestures. We are going to train the model with hand Gestures. Even we may use AI for Recognizing the clear image even with Bad background. We are providing the various features in Image viewing with Interaction module.
4.	Social Impact / Customer Satisfaction	Our model will help the doctors in OT via contactless interaction which gives Germ free communication, and it will reduce the sterilizing process in operation. Which ensures the infections to the patients.
5.	Business Model (Revenue Model)	We will provide our model for subscription-basedManner we could generate revenue though this method.
6.	Scalability of the Solution	In future we can expand our project via more additional gestures for browsing. Even we may implement the multiple inputs a time.

3.2Problem Solution fit

In this we prepared problem - solution fit document and submit for review.

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. 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 DESIGN

● 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 must complete all the activities and tasks listed below:

● Data Collection.

  ○ Collect the dataset or create the dataset

● Data Preprocessing

  ○ 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

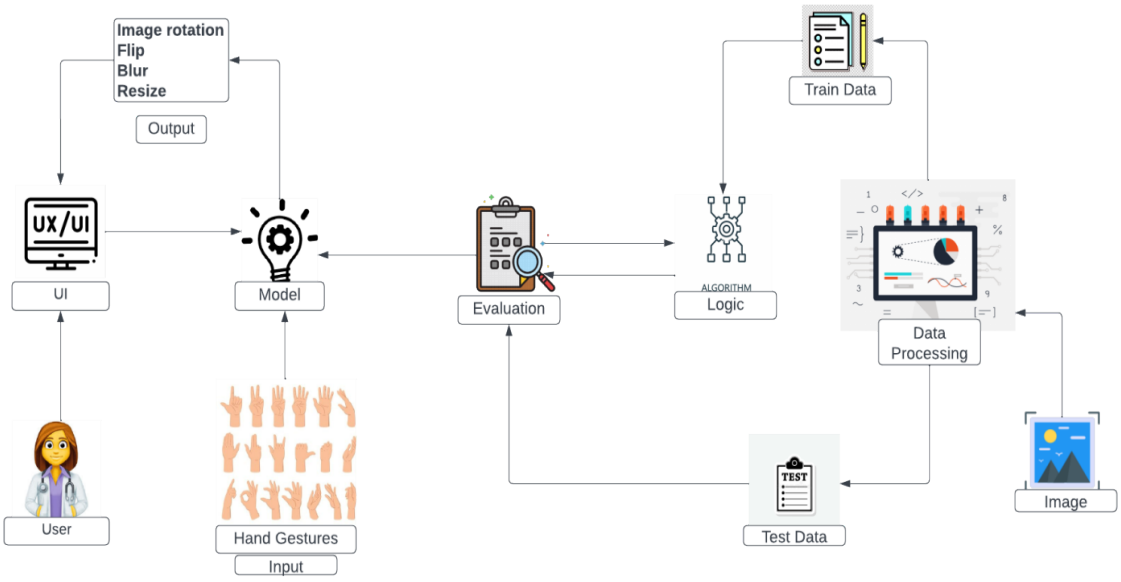
  ○ Configure the Learning Process

- 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
  - 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 DIAGRAM



7 COMPONENTS & TECHNOLOGIES

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App etc.	HTML, CSS, JavaScript ,React JS
2.	Application Logic-1(Data Preprocessing)	Variety of frameworks, libraries and Images are imported	Python
3.	Application Logic- 2(model building)	Build CNN model to convert the hand gestures to surf on the internet and communicate with computer.	Python,IBM Watson STT service
4.	Application Logic- 3 (Application Building)	Create HTML file for front end	HTML, CSS, Javascript .
5.	Dataset	Collect the hand gesture dataset	From Internet
6.	Cloud Database	Database service on cloud to train model	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage should be highly flexible, scalable to store the code and dataset.	IBM Block Storage or Local File system
8.	Machine Learning Model	Machine Learning Model deals withvarious algorithms that are needed for the implementation	CNN, Opencv, Object recognition model etc.
9.	Infrastructure	Application Deployment on Local System /Cloud Local Server Configuration: Install the windows version and execute the installer.	Local, Cloud Foundry, Kubernetes, etc.

8 SOURCE CODE

https://github.com/IBM-EPBL/IBM-Project-22383-1659850856/tree/main/PROJECT%20DEVELOPMENT%20PHASE

home.html

```
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
    <!--link rel="stylesheet" href="home.css"-->
    <style>
      body {
        background-image: url('https://tse2.mm.bing.net/th?id=OIP.0jyFTQx2umY6fRIB-q3FqAHaHO&pid=Api&P=0');
        background-repeat:  no-repeat;
        background-attachment: fixed;
        background-size: cover;
```

```
    }
    </style>
  <title>
    Gesture Recognition System
  </title>
</head>



<body>
<!--navigation bar-->
<nav class="navbar navbar-expand-md navbar-light">
 <div class="container-xxl">
  <a href="{{ url_for('home')}}" class="navbar-brand">
   <span class="fw-normal text-black">
    Gesture Recognition System
   </span>
  </a>
  <!--menu(toggle) button for mobile -->
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#main-nav" aria-controls="main-nav" aria-
expanded="false" aria-label="Toggle navigation">
   <span class="navbar-toggler-icon"></span>
  </button>
  <!--navbar links-->
  <div class="collapse navbar-collapse justify-content-end align-center"
id="main-nav">
   <ul class="navbar-nav">
    <li class="nav-item">
```

```
        <a href="{{ url_for('home')}}" class="nav-link active">Home</a>
      </li>
      <li class="nav-item">
        <a href="{{ url_for('intro')}}" class="nav-link
active">Introduction</a>
      </li>
      <li class="nav-item">
        <a href="{{ url_for('image1')}}" class="nav-link
active">Launch</a>
      </li>
    </ul>
   </div>
  </div>
</nav>

  <div class="container-lg m-3">
   <div class="row justify-content-center align-items-center">
    <div class="text-center text-md-center">
      <h1>
        <div class="display-2 text-dark">Hand Gesture Recognition</div>
        <div class="display-5 text-dark-50">For Sterile Browsing of
Radiology Images</div>
      </h1>
      <div class="text-center d-none d-md-block m-1">
        <img class="rounded mx-auto d-block"
src="https://cdn1.iconfinder.com/data/icons/hand-gestures-66/48/hand-
gesture-count-two-512.png" alt="gesture img">
      </div>
```

```
        <div class=" my-2 text-center">
          <a type="button" class="btn btn-primary btn-lg" href="{{
url_for('intro')}}">Click Here To Know More</a>
          <a type="button" class="btn btn-success btn-lg" href="{{
url_for('image1')}}">Launch</a>
          </button>
       </div>
     </div>
 <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.
min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V
8Qbsw3" crossorigin="anonymous"></script>
</body>
</html>
```

intro.html

```
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="C:\Users\Admin\Desktop\git\sprint-3\flask\static\New
folder\min1.css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv
1WTRi" crossorigin="anonymous">
    <!--link rel="stylesheet" href="home.css"-->
    <style>
      body {
```

```
        background-image:
url('https://tse2.mm.bing.net/th?id=OIP.0jyFTQx2umY6fRIB-
q3FqAHaHO&pid=Api&P=0');
        background-repeat:  no-repeat;
        background-attachment: fixed;
        background-size: cover;
       }
      </style>
    <title>


    </title>
</head>



<body>
<!--navigation bar-->
<nav class="navbar navbar-expand-md navbar-light">
  <div class="container-xxl">
    <a href="file:///C:/Users/Admin/Desktop/git/sprint-
3/flask/template/home.html" class="navbar-brand">
      <span class="fw-normal text-black">
        Gesture System
      </span>
    </a>
    <!--menu(toggle) button for mobile -->
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#main-nav" aria-controls="main-nav" aria-
expanded="false" aria-label="Toggle navigation">
```

```
      <span class="navbar-toggler-icon"></span>
    </button>
    <!--navbar links-->
    <div class="collapse navbar-collapse justify-content-end align-center" id="main-nav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a href="file:///C:/Users/Admin/Desktop/git/sprint-3/flask/template/home.html" class="nav-link active">Home</a>
        </li>
        <li class="nav-item">
          <a href="file:///C:/Users/Admin/Desktop/git/sprint-3/flask/template/intro.html" class="nav-link active">Introduction</a>
        </li>
        <li class="nav-item">
          <a href="file:///C:/Users/Admin/Desktop/git/sprint-3/flask/template/launch.html" class="nav-link active">Launch</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

  <div class="container-lg m-3">
    <div class="row justify-content-center align-items-center">
      <div class="text-center text-md-center">
        <h1>
          <div class="display-2 text-dark">Hand Gesture Recognition</div>
```

```
<div class="display-4 text-dark">About the System</div>

<br>

<small class="lead text-dark">

<h2>Hand gestures are a form of nonverbal communication that can be used in several fields such as communication between deaf-mute people, robot control, human–computer interaction (HCI), home automation and medical applications. Research papers based on hand gestures have adopted many different techniques, including those based on instrumented sensor technology and computer vision

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.

</h2></small>

</h1>

<div class="text-center d-none d-md-block m-1">

<img class="rounded mx-auto d-block" src="https://media.tenor.com/GwMnZWwxip0AAAAC/hand-finger-counting.gif" alt="gesture img">

</div>

<div class=" my-2 text-center">

<a type="button" class="btn btn-primary btn-lg" href="file:///C:/Users/Admin/Desktop/git/sprint-3/flask/template/intro.html">Click Here To Know More</a>

<a type="button" class="btn btn-success btn-lg" href="file:///C:/Users/Admin/Desktop/git/sprint-3/flask/template/launch.html">Launch</a>

</button>

</div>

</div>
```

```
   <script src="C:\Users\Admin\Desktop\git\sprint-3\flask\static\New folder\min1.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>

</body>
</html>
```

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="C:\Users\Admin\Desktop\git\sprint-3\flask\static\New folder\min.js"></script>

    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <meta charset="UTF-8">

    <title>Predict</title>

    <link href="C:\Users\Admin\Desktop\git\sprint-3\flask\static\New folder\min.css" rel="stylesheet">
```

```
    <link href="C:\Users\Admin\Desktop\git\sprint-
3\flask\static\css\main.css" rel="stylesheet">
    <style>
      body {
        background-image:
url('https://media.npr.org/assets/img/2018/08/16/gettyimages-
507850475_wide-
e729499092c4106cff752cea8a2c49623c1595d3.jpg?s=1400');
        background-repeat:  no-repeat;
        background-attachment: fixed;
        background-size: cover;
      }
      </style>
  </head>

  <body>
  <!--navigation bar-->
    <nav class="navbar navbar-expand-md navbar-light">
  <div class="container-xxl">
    <a href="file:///C:/Users/Admin/Desktop/git/sprint-
3/flask/template/home.html" class="navbar-brand">
      <span class="fw-normal text-black">
        Gesture Recognition System
      </span>
    </a>
    <!--menu(toggle) button for mobile -->
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#main-nav" aria-controls="main-nav" aria-
expanded="false" aria-label="Toggle navigation">
```

```
      <span class="navbar-toggler-icon"></span>
    </button>
    <!--navbar links-->
    <div class="collapse navbar-collapse justify-content-end align-center" id="main-nav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a href="file:///C:/Users/Admin/Desktop/git/sprint-3/flask/template/home.html" class="nav-link active">Home</a>
        </li>
        <li class="nav-item">
          <a href="file:///C:/Users/Admin/Desktop/git/sprint-3/flask/template/intro.html" class="nav-link active">Introduction</a>
        </li>
        <li class="nav-item">
          <a href="file:///C:/Users/Admin/Desktop/git/sprint-3/flask/template/launch.html" class="nav-link active">Launch</a>
        </li>
      </ul>
    </div>
  </div>
    </nav>
    <br>

    <div1 style="text-align: center;"><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>Provide 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...
    </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="C:\Users\Admin\Desktop\git\sprint-
3\flask\static\js\main.js" type="text/javascript"></script>

  </footer>
  </html>
```

app.py

```
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

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

import numpy as np


from tensorflow.keras.models import load_model#to load our trained model

import os

from werkzeug.utils import secure_filename


app = Flask(__name__,template_folder="templates") # initializing a flask
app

# Loading the model

model=load_model('gesture.h5')

print("Loaded model from disk")
```

```
@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("launch.html")


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

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

    # 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)
    # Batch of 1
    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)
        cv2.imshow("Frame", frame)


        #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, (21, 21), 0)
        cv2.imshow("Blurred", blurred)
        key=cv2.waitKey(3000)
        if (key & 0xFF) == ord("3"):
            cv2.destroyWindow("Blurred")

    elif prediction[0][0]=='FOUR':
```

```
            resized = cv2.resize(image1, (400, 400))
            cv2.imshow("Fixed Resizing", resized)
            key=cv2.waitKey(3000)
            if (key & 0xFF) == ord("4"):
                cv2.destroyWindow("Fixed Resizing")


        elif prediction[0][0]=='FIVE':
            '''(h, w, d) = image1.shape
            center = (w // 2, h // 2)
            M = cv2.getRotationMatrix2D(center, 45, 1.0)
            rotated = cv2.warpAffine(image1, M, (w, h))'''
            gray = cv2.cvtColor(image1, cv2.COLOR_RGB2GRAY)
            cv2.imshow("OpenCV Gray Scale", gray)
            key=cv2.waitKey(3000)
            if (key & 0xFF) == ord("5"):
                cv2.destroyWindow("OpenCV Gray Scale")
            else:
            continue


        interrupt = cv2.waitKey(10)

        if interrupt & 0xFF == 27: # esc key
            break
    cap.release()cv2.destroyAllWindows()

  return render_template("home.html")
if__name__== "__main__":
  # running the app
  app.run(debug=True)
```

9 TESTING:

Test Cases

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	1	0	0	1
Client Application	1	0	0	1

User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the AI-based A Gesture-based Tool for Sterile Browsing of Radiology Image project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

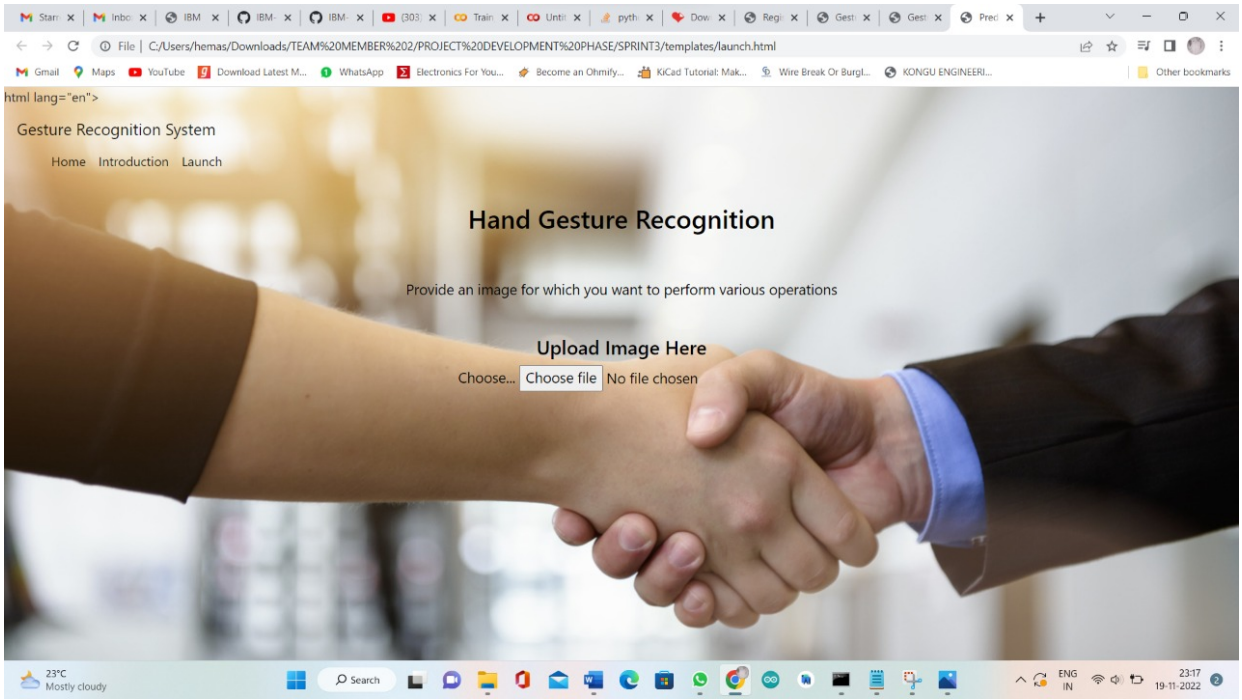
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	0	0	1
Duplicate	4	1	3	0	8
External	1	3	0	0	4
Fixed	2	4	4	2	12
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	8	8	4	2	22

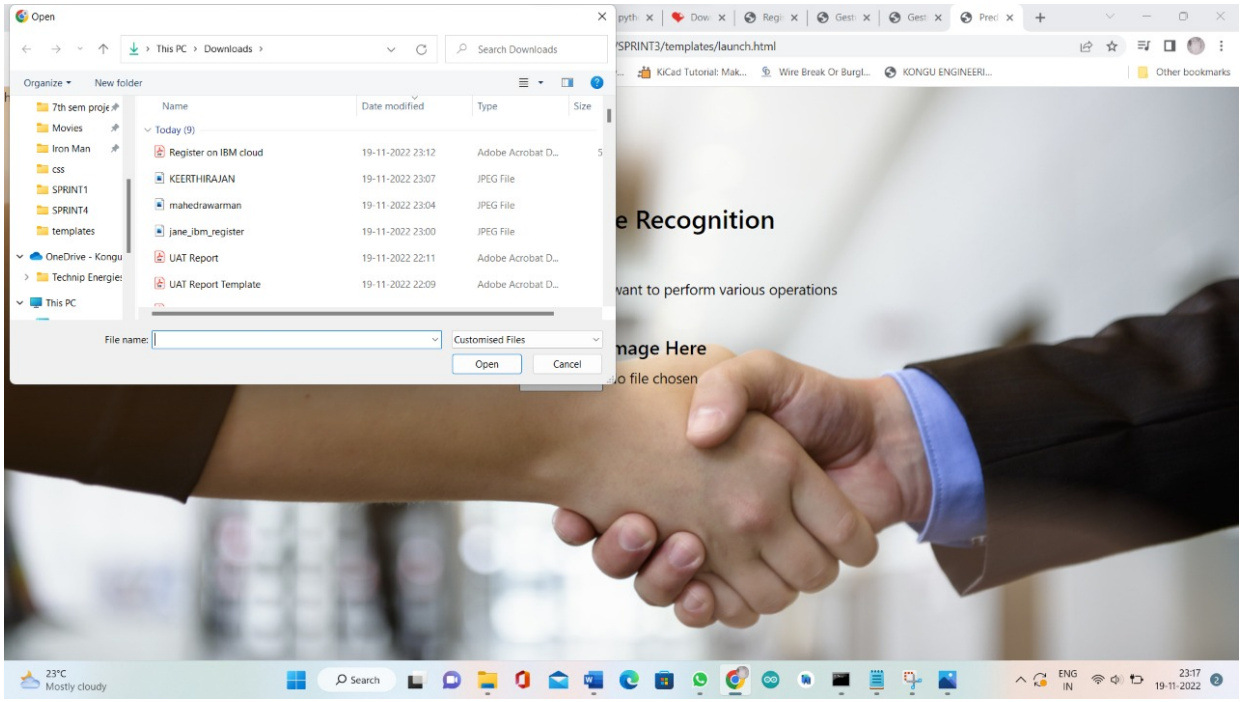
10 OUTPUT

10.1 HOME PAGE:

LAUNCH PAGE:

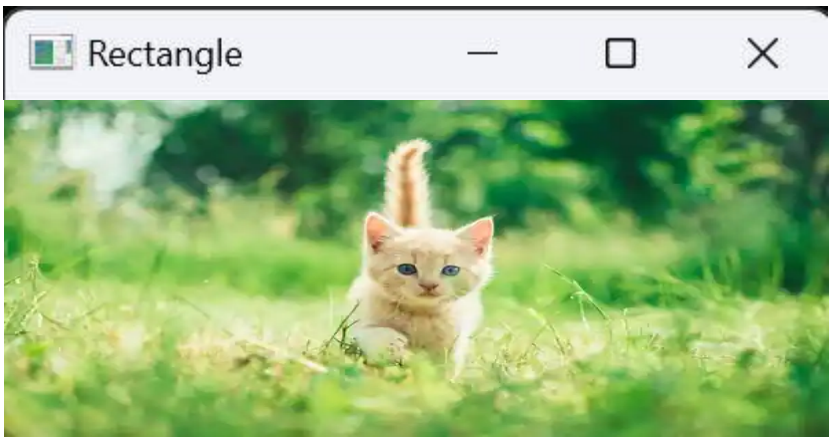


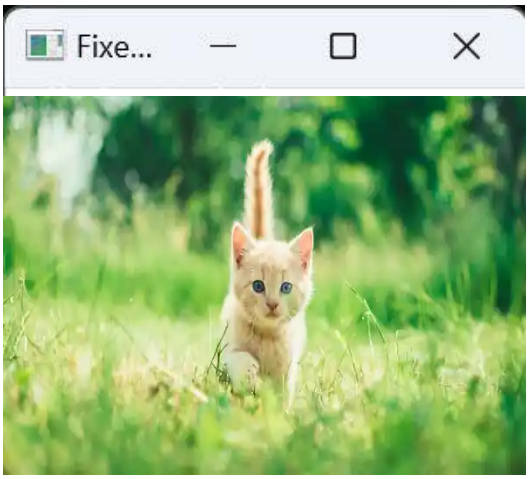
OPTION CHOOSING PAGE

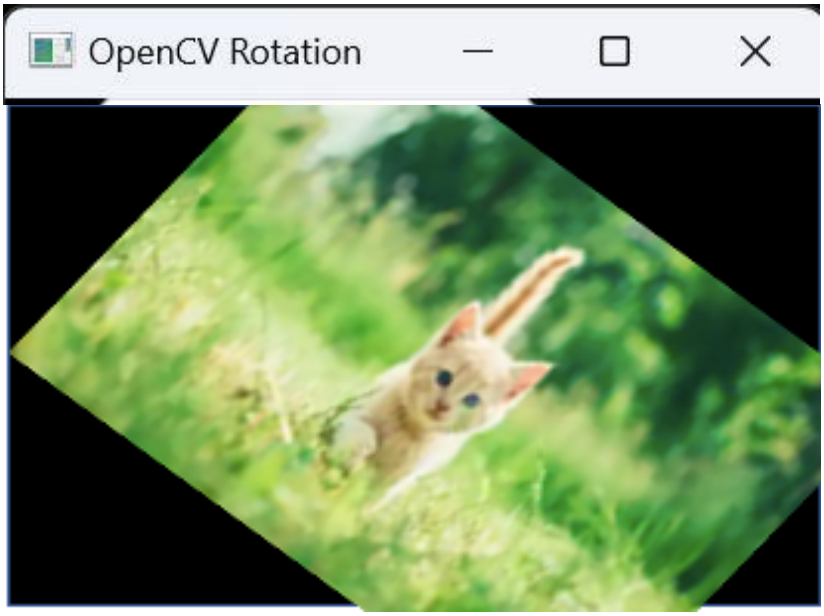


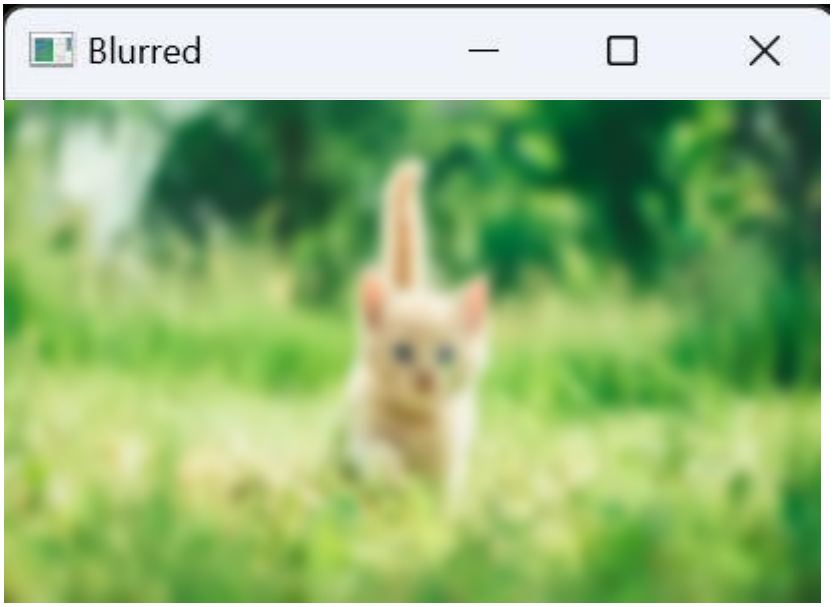
10.1 Actions:

- 0-Rectangle
- 1-Fixed Resizing (200,200)
- 2-OpenCV Rotation
- 3-Blurred
- 4-Fixed Resizing (400,400)
- 5-OpenCV Gray Scale




Fixe...


OpenCV Rotation


Blurred

11 RESULT

Final findings of the project along with code. Through this project we found that we can maintain the sterility of an operation theater, etc by using hand based gesture tools to browse the images

12 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.

13 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 teaching

14 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.

15 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 usability 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.

16 REFERENCE

1. Schultz M, Gill J, Zubairi S, Huber R, Gordin F. "Bacterial contamination of computer keyboards in a teaching hospital," *Infect Control Hosp. Epidemiol* 2003

2. 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

3. 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.

4. Wachs JP, Stern HI, Edan Y, et al. "Real-Time Hand Gesture Interface for Browsing Medical Images" *Int. J Intel. Comp. Med. Sci. Image Proc* 2007

17 GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-22383-1659850856

https://drive.google.com/drive/folders/102BGn-XXifdvVRfbm2s48680S-SNtucq?usp=sharing