**PROJECT REPORT**

**A GESTURE- BASED TOOL FOR STERILE**

**BROWSING OF**

**RADIOLOGY IMAGE**

**TEAM ID: PNT2022TMID50430**

**A Project Report**

**Submitted By**

**SIVASUBRAMANIAN.D 952319205041**

**MOHAMED AL SUBAIR.S 952319205027**

**SANTHOSH.A 952319205038**

**PRAVEENKUMAR.K 952319205031**

**CONTENT**

**1. INTRODUCTION**

1.1 Project Overview

1.2 Purpose

**2. LITERATURE SURVEY**

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

**3. IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

**4. REQUIREMENT ANALYSIS**

4.1 Functional requirement

4.2 Non-Functional requirements

**5. PROJECT DESIGN**

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

**6. PROJECT PLANNING & SCHEDULING**

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

**7. CODING & SOLUTIONING** (Explain the features added in the project along with code)

**8. RESULTS**

**9. ADVANTAGES & DISADVANTAGES**

**10. CONCLUSION**

**11. FUTURE SCOPE**

**12. APPENDIX**

Source Code

GitHub&Project Demo Link

# 

# A GESTURE- BASED TOOL FOR STERILE BROWSING

# OF RADIOLOGY IMAGE

**1 Introduction**

1.1 Project Overview:-

Recently, strong efforts have been carried out to develop intelligent and natural interfaces between users and computer-based systems based on human gestures. Gestures provide an intuitive interface to both humans and computers. Thus, such gesture-based interfaces can not only substitute the common interface devices but can also be exploited to extend their functionality.

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.

# 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

.

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 - 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.

# Literature Survey

2.1 Existing problem:

It 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, us of attention, and providing fast response times. This project,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.

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. 1 In this paper, we suggest the use of hand gestures as an alternative to existing interface techniques, offering the major advantage of sterility. Even though voice control also provides sterility, the noise level in the operating room (OR) deems it problematic.

 Surgeons kept their focus of attention between the patient and the surgical point of interest on the touch-screen navigation system.

 A short distance between the surgeon and the patient was maintained during most of the surgery.

 The surgeon had to move close to the main control wall to discuss and browse through the patient's MRI images.

**2.2 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;4(24):302-303. [PubMed] [Google Scholar]

2. Nishikawa A, Hosoi T, Koara K, Ngoro 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. [Google Scholar]

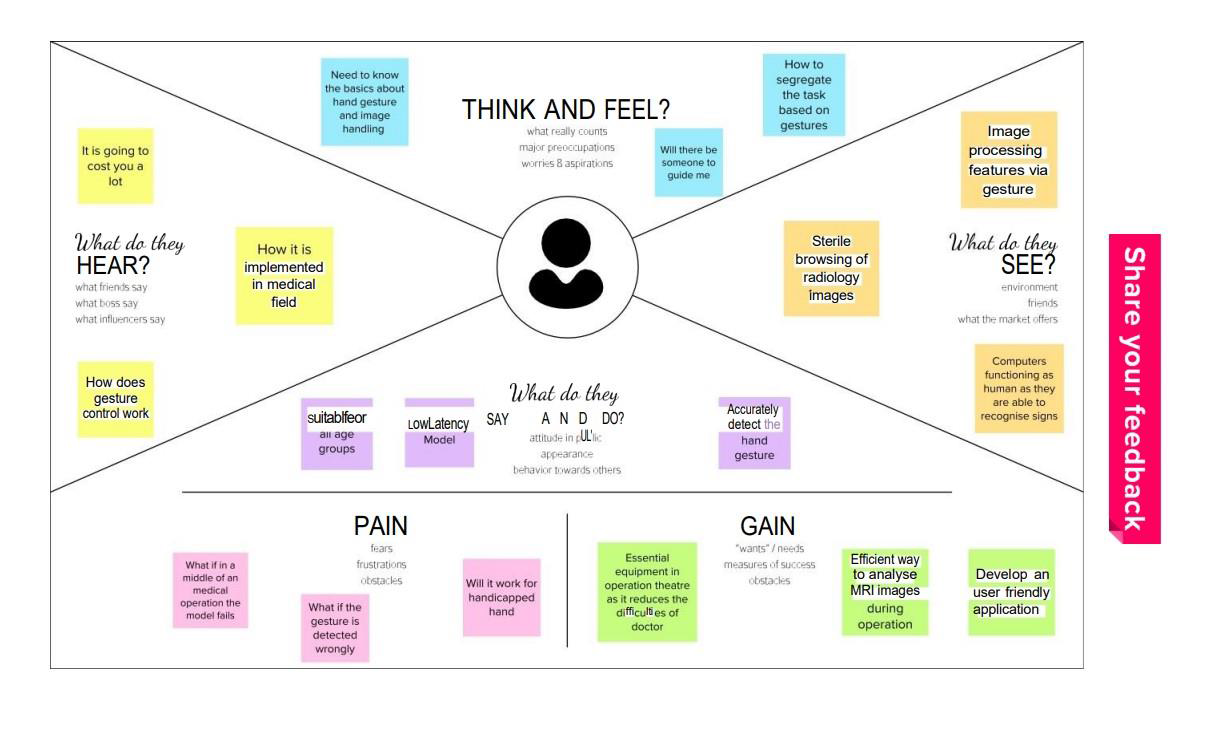
3. Smith KR, Frank KJ, Bucholz RD. “The NeuroStation- a highly accurate, minimally invasive solution to frameless stereotatic neurosurgery,” Comput Med Imaging Graph 1994;18:247-256. [PubMed] [Google Scholar]

**2.3 Problem Statement definition:**

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. Even though voice control also provides sterility, the noise level in the operating room (OR) deems it problematic.

# IDEATION & PROPOSED **SOLUTION**

**3.1 Empathy Map:**



**3.2 Ideation & Brainstorming:**

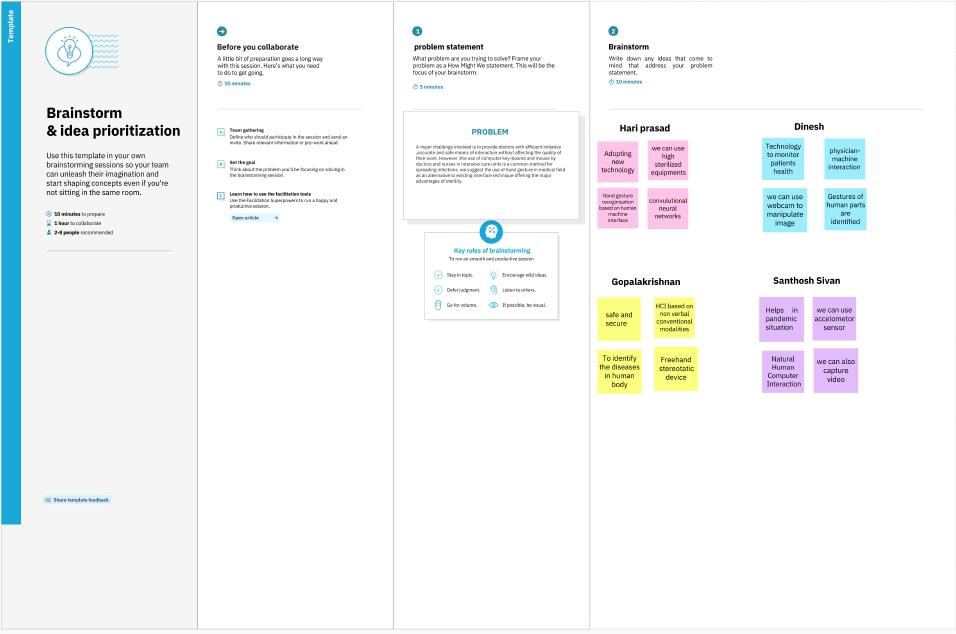
 In order to provide surgeons with a more efficient, comfortable, precise, and sterile interaction technique, the feet and hands can be an effective means of accomplishing this goal in comparison to other modalities, such as voice or gaze interaction or using Radar Sensor.

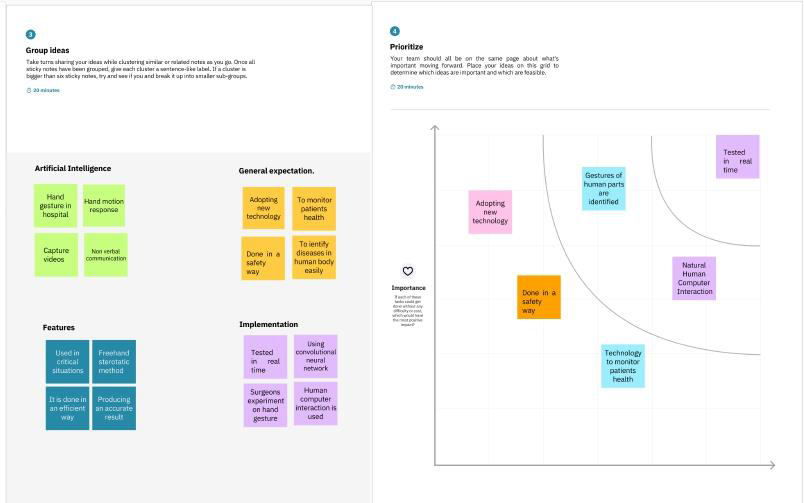
 Touch-less gesture interaction is an option to interact with imaging systems, displays, and controllers without breaking the sterility barrier. The system utilizes nothing but a camera with good quality and can follow the hand of the user in 2 dimensions and identify up to four mouse-defined hand motions.

 Recent progress in gaming technologies provides innovative opportunities for motion tracking and human-machine interaction. In the field of healthcare, sensors like Microsoft® Kinect (2015) have been used for detecting postures.

Using electromyography technology to capture gesture instead of the camera, therefore it is less affected by the external factors such as light and obstruction. The disadvantage of high computational cost.

 Voice command is another type of touchless communication but its commands are discrete rather than hand gestures which are able to perform analog commands. On the other hand, voice command has other disadvantages such as its low accuracy due to existence of noise in surgery rooms.

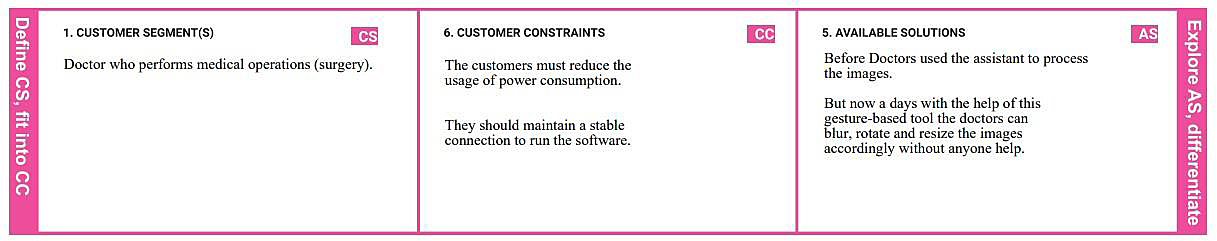


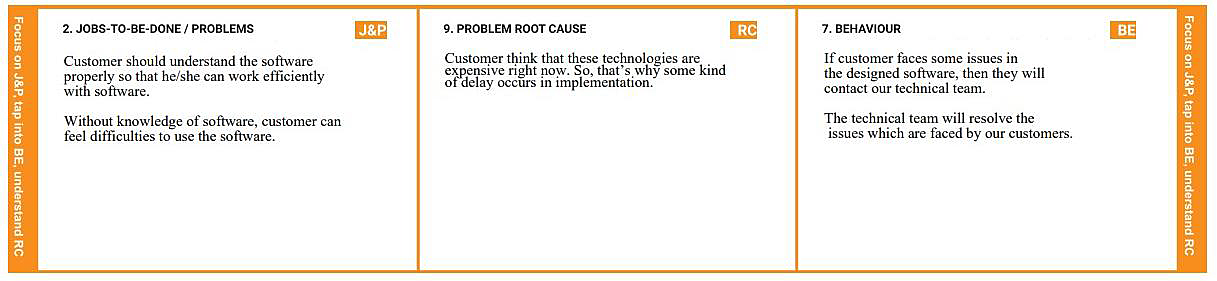


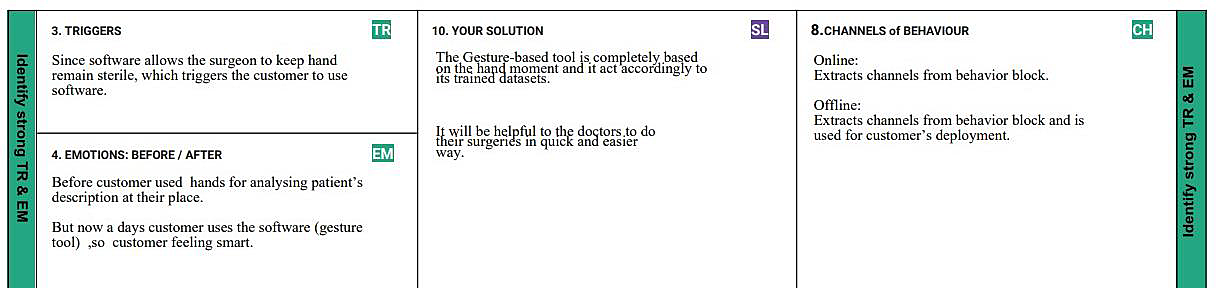
**3.3 Proposed Solution:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **PARAMETER** | **DESCRIPTION** |
| 1 | Problem Statement (Problem to be solved) | Hand Gesture tool to do Contactless navigation of radiology images to simplify the communication of  Doctors with machines. |
| 2 | Idea / Solution description | The technology used to assist doctors by taking hand gestures as input and perform necessary  actions in machines. |
| 3 | Novelty / Uniqueness | These Gesture helps us to visualize the Words and help Gain the Listener’s Attention. |
| 4 | Social Impact / Customer Satisfaction | The proposed system should maintain a good balance between complexity, accuracy and applicability. |
| 5 | Business Model (Revenue Model) | The revenue model is based on the integration of this technology in radiology devices the business will be either tie up with companies or specific sale of machines. |
| 6 | Scalability of the Solution | The proposed approach allows the learning of new gestures with no need of recording real subjects. |

**3.4 Problem Solution fit:**







**REQUIREMENT ANALYSIS**

**4.1. Functional requirement:**

**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:**

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

|  |  |  |
| --- | --- | --- |
| **FR NO** | **Non-Functional Requirements** | **Description** |
| FR-1 | Usabili | 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 |
| FR-2 | Security | This system is protected and only authorized users can access it |
| FR-3 | Relaiblility | After installing the Application,the system will predict the gesture and performs sterile browsing |
| FR-4 | Performance | The system responds to a user in seconds and the hardware and software works well |
| FR-5 | Availability | It is accessible by authorised user from anywhere at any time whenever there is an emergency |
| FR-6 | Scalability | This system allows more number of users at a time  and there is no loss can be identified |

**PROJECT DESIGN**

**5.1. Data Flow Diagrams:**

● 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:**

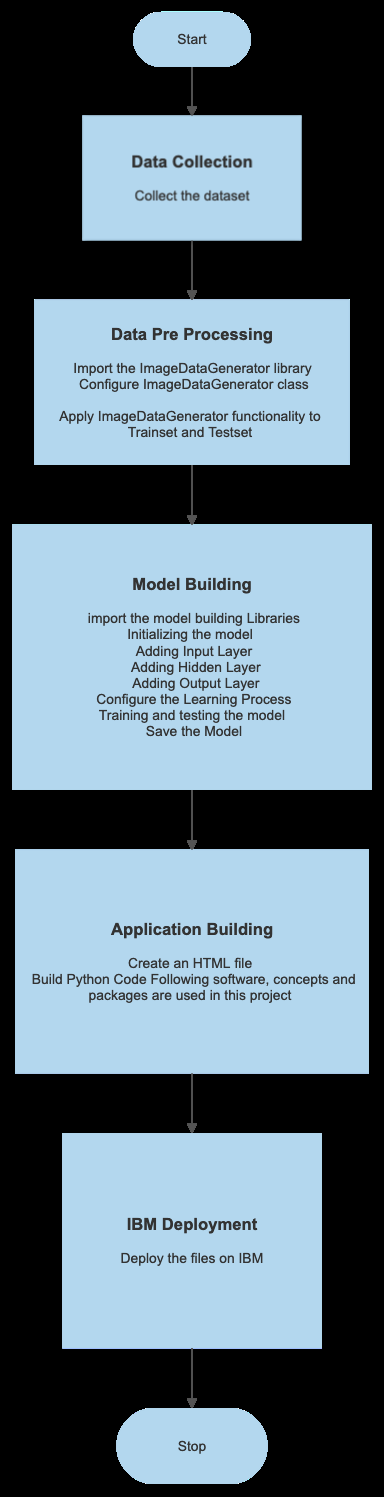
○ open anaconda prompt as administrator

○ Type “pip install Tensor Flow” (make sure you are working on python 64 bit)

○ Type “pip install opencv-python”

○ Type “pip install flask

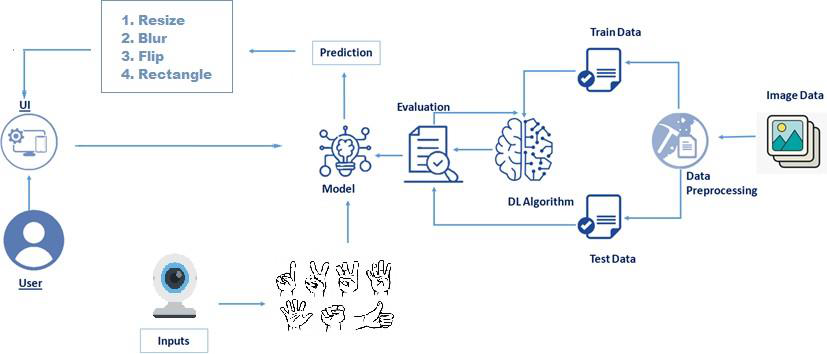
# FLOW CHART:



**5.2. Solution & Technical Architecture:**

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.

**Architecture:**



**User Stories:**

It 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.

**PROJECT PLANNING & SCHEDULING**

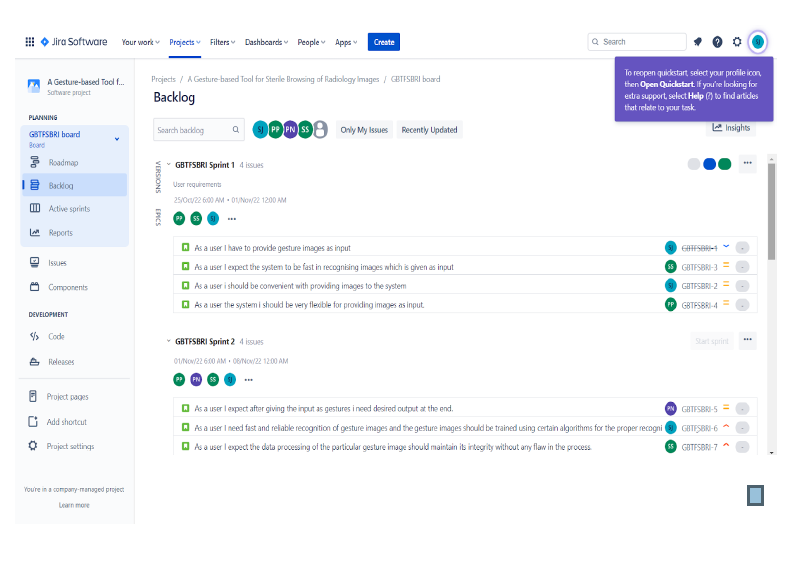
**6.1. Sprint Planning & Estimation:**

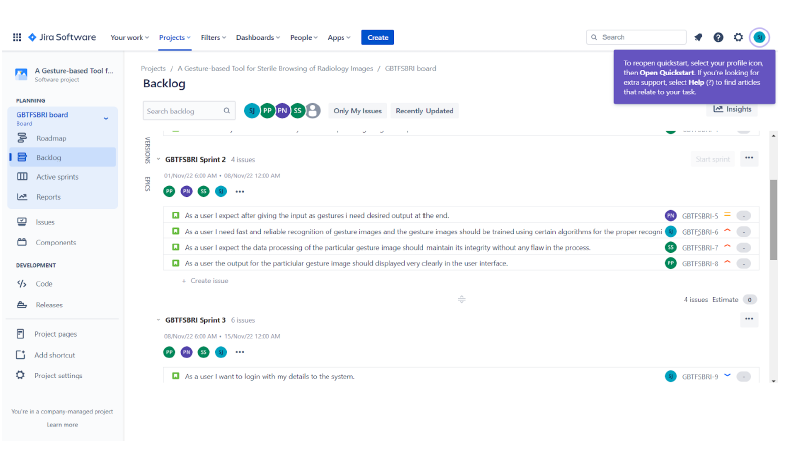
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement(Epic)** | **User Story number** | **UserStory/Task** | **Story Points** | **Priority** | **ReamMembers** |
| Sprint-1 | Launching Software | USN-1 | As a user, I can launch the developed software | 1 | Low | Sivasubramanian  Santhosh  Mohamad Al Subair  Praveenkumar |
| Sprint-1 | AccessUI | USN-2 | As a user, I will use the software and operate on the UI | 1 | Medium | Sivasubramanian  Santhosh  Mohamad Al Subair  Praveenkumar |
| Sprint-2 | LaunchingCamera | USN-3 | As a user, I can open the camera from the software to perform gesture | 1 | Low | Sivasubramanian  Santhosh  Mohamad Al Subair  Praveenkumar |
| Sprint-2 | Upload images fromlocalsystem | USN-4 | As a user, I can upload images to the software from the local system | 2 | Low | Sivasubramanian  Santhosh  Mohamad Al Subair  Praveenkumar |
| Sprint | Functional Requirement(Epic) | User storyNumber | User Story / Task | Story Points | Priority | Sivasubramanian  Santhosh  Mohamad Al Subair  Praveenkumar |
| Sprint-3 | Perform Gesture | USN-5 | As a user, I can perform various gesture with respect to system specification for processing | 2 | Medium | Sivasubramanian  Santhosh  Mohamad Al Subair  Praveenkumar |
| Sprint-4 | Output | USN-6 | As a user. I can see the sterile browsers image with respect to the gesture performed, display on the screen | 2 | High | Sivasubramanian  Santhosh  Mohamad Al Subair  Praveenkumar |

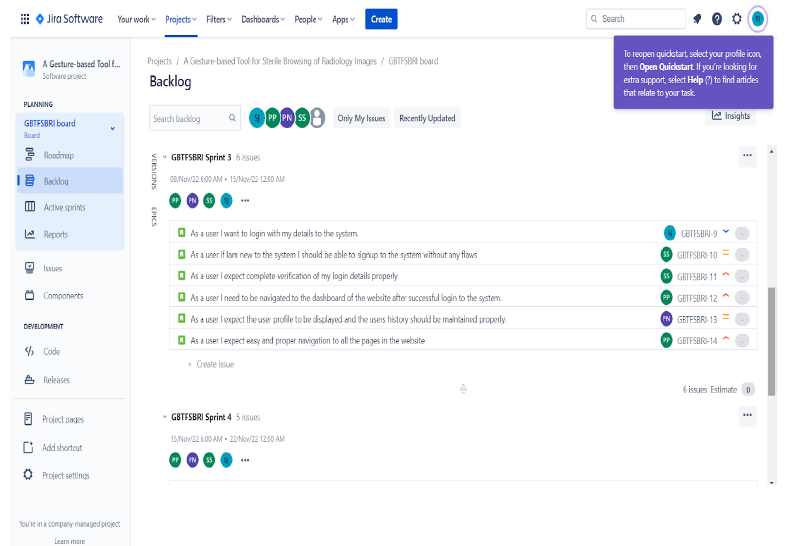
**6.2. Sprint Delivery Schedule:**

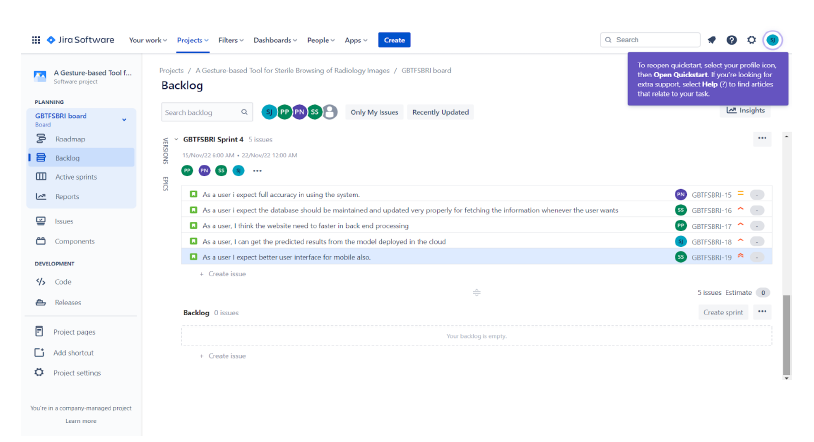
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint EndDate (Planned)** | **Story Points Completed** | **Sprint Release Date(Actual)** |
| Sprint-1 | 20 | 24Oct2022 | 29Oct2022 | 29Oct2022 | 20 | 30Oct2022 |
| Sprint-2 | 20 | 31Oct2022 | 05Nov2022 | 05Nov2022 | 20 | 06Nov2022 |
| Sprint-3 | 20 | 07Nov2022 | 12Nov2022 | 12Nov2022 | 20 | 15Nov2022 |
| Sprint-4 | 20 | 14Nov2022 | 19Nov2022 | 19Nov2022 | 20 | 19Nov2022 |

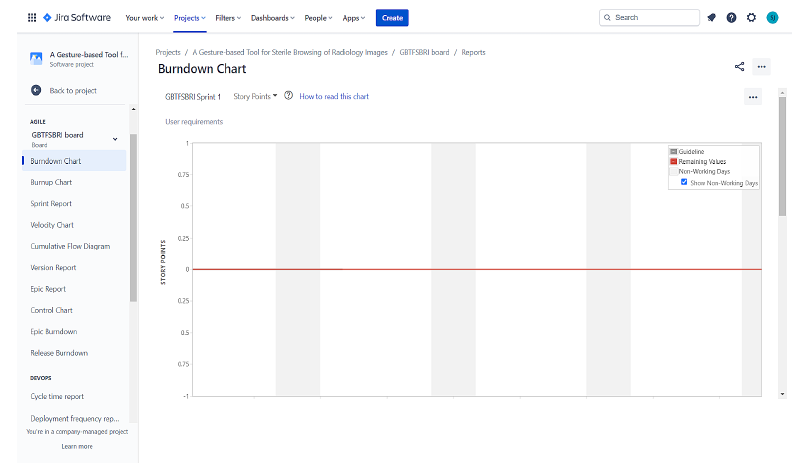
**6.3 REPORTS FROM JIRA :**











**7 CODING & SOLUTIONING**

**7.1 FEATURE 1 :**

● Here we are not use any kind of difficult Process like user login and other stuffs because these applications are used only in operation theatre and used by Doctors.

● First the user must know about the application, so we have giving some brief explanation in the application in Introduction page.

● In the home page itself they can upload the gesture image and get the desired output.

**7.2 FEATURE 2 :**

● Our Next feature is about the images which going to be show in

different angles and stages.

● By uploading the radiology image, we easily change the Images in different angles and Different filters like blurring the image for better visible of the parts of human body.

**8.TESTING**

**8.1 TEST CASES :**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Feature type** | **Component** | **Test Scenario** | **Pre-Requiste** | **Steps to execute** | **Expected**  **Result** | **Actual**  **Result** | **Status** | **Comments** | **Expected by** |
| Functional | Home page | Verify user is  able to see the login/signup popup when user clicked on my account button |  | 1.Enter URLand click go  2.Click on myAccount dropdown button  3.Verify login/Signup popup displayed or not | Login/signup popup should display | Working as expected | Pass |  | SIVASUBRAMANIAN.  MOHAMAD AL SUBAIR.S  SANTHOSH.A  PRAVEENKUMAR.K |
| UI | Home Page | Verify the UI elements in Login/signup popup |  | 1.Enter URL and clickgo  2.Click on myAccount dropdown button  3.Verify login/Signup popup with below UL element | Application should show below UI elements | Working as expected | Pass |  | SIVASUBRAMANIAN.  MOHAMAD AL SUBAIR.S  SANTHOSH.A  PRAVEENKUMAR.K |
| Functional | Home page | Verify user is able to login into application with Valid credential |  | 1.Enter URL(https;//shopenzer.com)and click go  2.Click on myAccount dropdown button  3.enter Valid username/email in Email textbox  4.Enter valid password in password textbox  5.Click on login button | User should navigate to to user account homepage | Working as expected | Pass |  | SIVASUBRAMANIAN.  MOHAMAD AL SUBAIR.S  SANTHOSH.A  PRAVEENKUMAR.K |
| Functional | Login page | Verify is able to login into application with INvalid credentials |  | 1.Enter URL(https;//shopenzer.com)and click go  2.Click on myAccount dropdown button  3.enter Valid username/email in Email textbox  4.Enter valid password in password textbox  5.Click on login button | Application should show "incorrect email or password" validation message | Working as expected | Pass |  | SIVASUBRAMANIAN.  MOHAMAD AL SUBAIR.S  SANTHOSH.A  PRAVEENKUMAR.K |
| Functional | Login page | Verify is able to login into application with Invalid credentials |  | 1.Enter URL(https;//shopenzer.com)and click go  2.Click on myAccount dropdown button  3.enter Valid username/email in Email textbox  4.Enter valid password in password textbox  5.Click on login button | Application should show "incorrect email or password" validation message | Working as expected | Pass |  | SIVASUBRAMANIAN.  MOHAMAD AL SUBAIR.S  SANTHOSH.A  PRAVEENKUMAR.K |
| Functional | Login Page | Verify is able to login into application with Invalid credentials |  | 1.Enter URL(https;//shopenzer.com)and click go  2.Click on myAccount dropdown button  3.enter Valid username/email in Email textbox  4.Enter valid password in password textbox  5.Click on login button | Application should show "incorrect email or password" validation message | Working as expected | Pss |  | SIVASUBRAMANIAN.  MOHAMAD AL SUBAIR.S  SANTHOSH.A  PRAVEENKUMAR.K |

**8.2 USER ACCEPTANCE TESTING :**

The purpose of this document is to briefly explain the test

coverage and open issues of the A gesture based tool for sterile

browsing of radiology images project at the time of the release to User

Acceptance Testing (UAT).

**DEFECT ANALYSIS:**

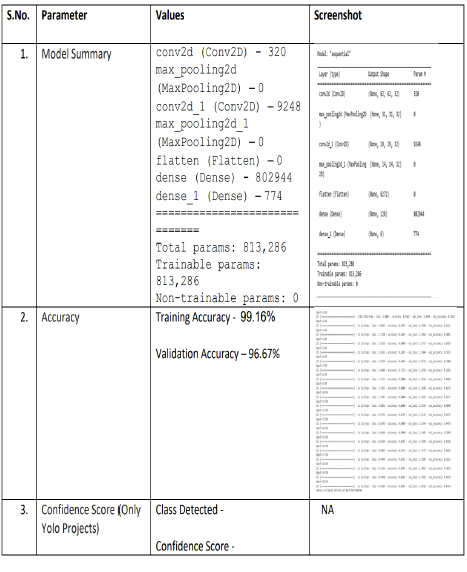
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Sub Total** |
| By Design | 5 | 4 | 2 | 1 | 12 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 3 | 0 | 1 | 6 |
| Fixed | 5 | 2 | 4 | 1 | 12 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 2 | 0 | 1 | 1 | 4 |
| Won't Fix | 1 | 2 | 2 | 1 | 7 |
| Totals | 16 | 11 | 13 | 5 | 46 |

**TEST CASE ANALYSIS:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Sub Total** |
| By Design | 5 | 4 | 2 | 1 | 12 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 3 | 1 | 1 | 6 |
| Fixed | 5 | 2 | 1 | 1 | 12 |
| Not Reproduced | 0 | 0 | 0 | 0 | 1 |
| Skipped | 2 | 0 | 1 | 1 | 4 |
| Won't Fix | 1 | 2 | 1 | 1 | 7 |
| Totals | 16 | 11 | 5 | 5 | 46 |

**9.RESULTS:**

**9.1 PERFORMANCE METRICS:**

W

**10.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 recognize or process it.

**11.CONCLUSION:**

As the technology is booming with emerging trends therefore the

virtual I/O interface which can possibly contribute to public healthcare. We used OpenCV and TensorFlow to detect movements and gestures. The

models were trained with hand images.

The accuracy of the model is achieved and the optimization of the

model is a continuous process and we are building an accurate solution by

tuning the hyper parameters.

This specific model could be used as a use case for Virtual Human-

Computer Interaction. By developing this system, we can allow the user to

have a contact-less interactive experience, which would be of great help to society.

In this project we developed a tool which recognizes 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 user's location since they don’t have to be in contact with any device. It alsodoesnot require the user to have any device on them to use it.

**12. FUTURE SCOPE :**

The tool can be made quicker by increasing the recognition speed.

More and 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.

**13.APPENDIX:**

Sorce code:

form flask import Flask,render\_template,request

# Flask-It is our framework which we are going to use to run/serve our applicatin.

#request-for accessing file which was uploaded by user on our applicatin.

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="template")#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 inteo page

@app.route('/image1',methods=['GET','POST'])# routes to the index html

def image1():

return render\_template("launch.html")

@app.route('/predict',mrthods=['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.jion(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])

x1 = 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.thershold(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)

#Displaing 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 predicton[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"):

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()

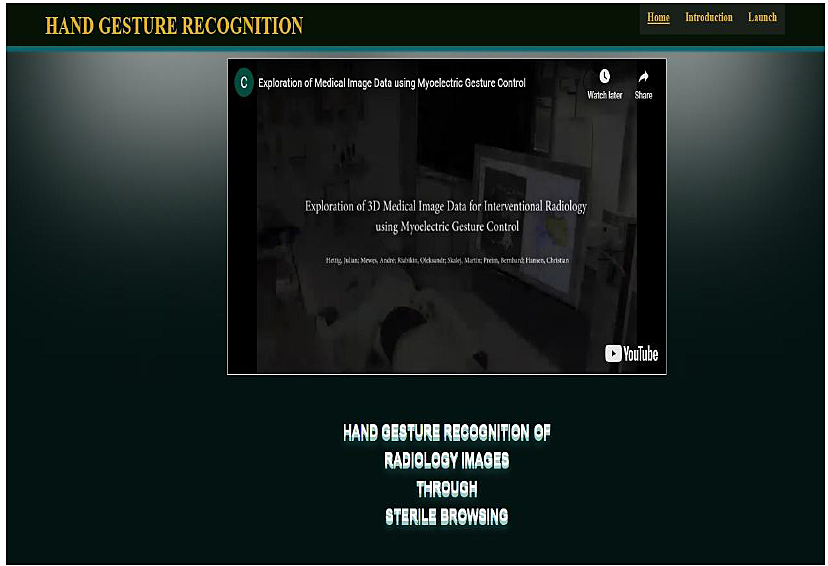
return render\_template("home.html")

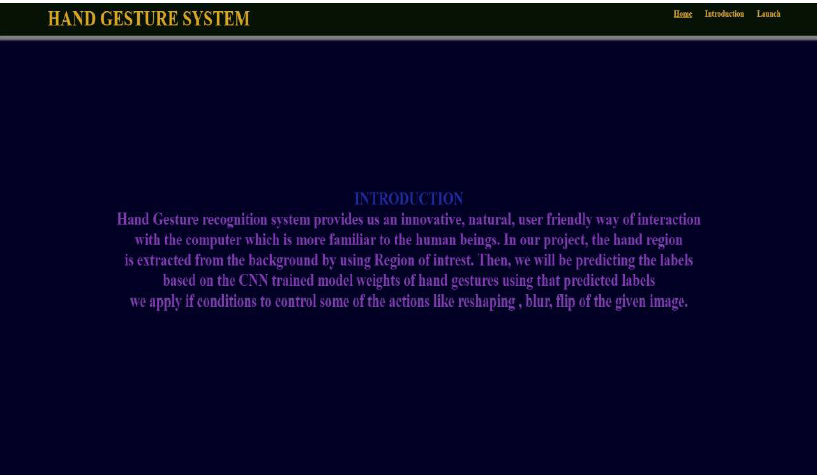
if\_\_name\_\_== "\_\_main\_\_":

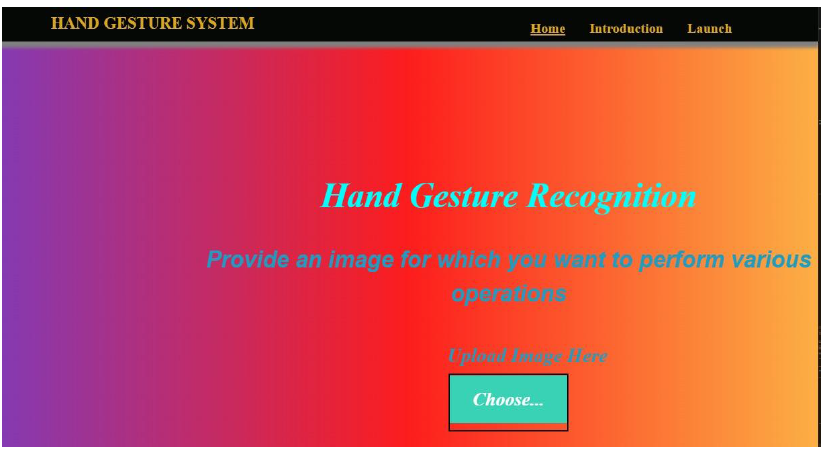
# running the app

app.run(debug=False)

**SCREENSHOT OF WEB APP:**







**GitHub Link:**

**<https://github.com/IBM-EPBL/IBM-Project-50044-1660890073>**

**Demo Link:**

**<https://youtu.be/70FWDLSYsA0>**