A Gesture-based Tool for Sterile Browsing of Radiology Images

TEAM ID: PNT2022TMID28899

1. INTRODUCTION

1.1 Project Overview

Humans are able to 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 predicted is 1 then images is blurred;2, image is resized;3,image is rotated etc.

1.2 Purpose

Gesture Recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current focus in this paper is Hand Gesture Recognition. Computer recognition of hand gestures may provide a more natural human computer interface, allowing people to point, or rotate a CAD model by rotating their hands. Interactive computer games would be enhanced if the computer could understand players' hand gestures. Gesture recognition may even be useful to control household appliances. We distinguish two categories of gestures: static and dynamic. A static gesture is a particular hand configuration and pose, represented by a single image. A dynamic gesture is a moving gesture, represented by a sequence of images. We focus on the recognition of static gestures, although our method generalizes in a natural way to dynamic gestures.

2. LITERATURE SURVEY

2.1 Existing Problem

The current medical infrastructure largely uses human interaction via mouse, keyboard and other conventional input devices. These human computer interactions are major reason for the transfer of illness in operation rooms in the hospital. Doctors in intensive care units (ICU) frequently transfer illnesses by using computer keyboards and mice. The sterile environment is often compromised leading to the spread of diseases.

2.2 References

1. Java 2 Complete reference Fifth Edition – Herbert Scheldt.
2. Enabling Cursor Control Using on Pinch Gesture Recognition - Benjamin Baldus, Debra Lauterbach, Juan Lizarraga, October 5, 2007.
3. Pro javatm 6 3D Game Development – Andrew Davison [2008]
4. Enabling Cursor Control Using on pinch gesture Recognition Benjamin Baldus, Debra Lauterbach, Juan Lizarraga October 5, 2007
5. "A Fast Algorithm For Vision-Based Hand Gesture Recognition For Robot Control" Asanterabi
6. Malima, Erol Özgür, and Müjdat Çetin Faculty of Engineering and Natural Sciences, Sabancı University, Tuzla, İstanbul, Turkey.
7. Orientation Histograms for Hand Gesture Recognition William T. Freeman, Roth.Michal
8. Gesture interface device developed by Ben-Gurion University of the Negev Eureka! Science News.htm

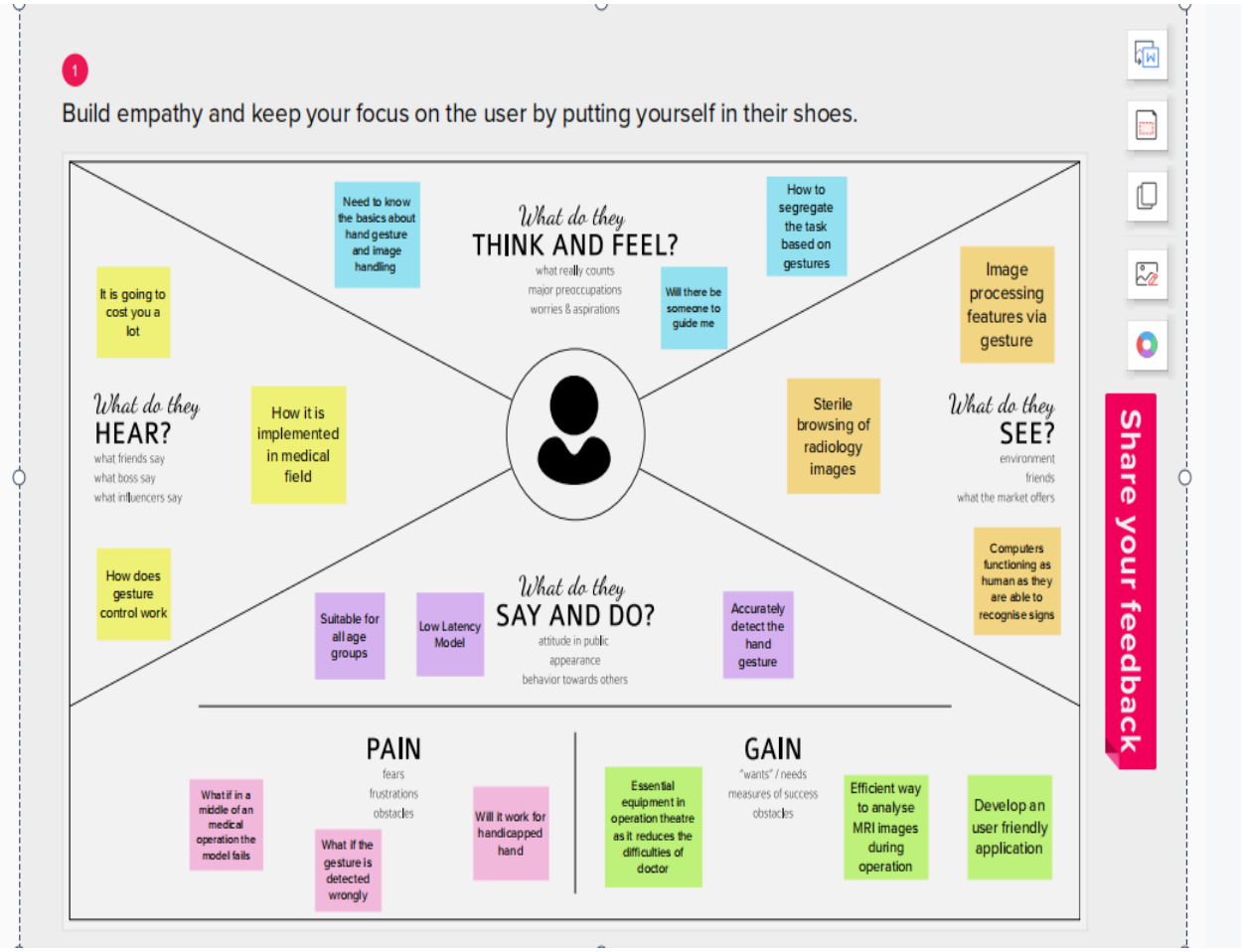
2.3 Problem Statement Definition

Humans are able to 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.

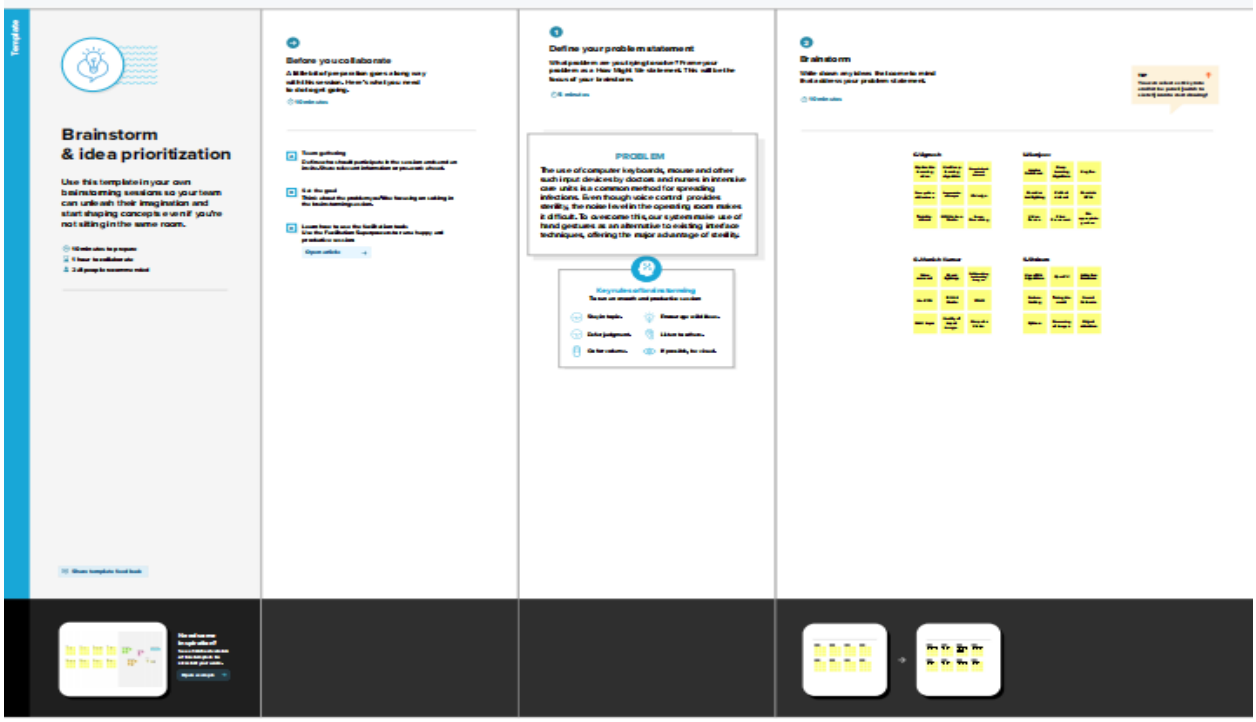
In this project Gesture based Desktop automation ,First the model is trained pre trained on the images of different hand gestures, such as a showing numbers with fingers as 1 ,2,3,4 . This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the gesture predictes is 1 then images is blurred;2, image is resized;3,image is rotated etc.

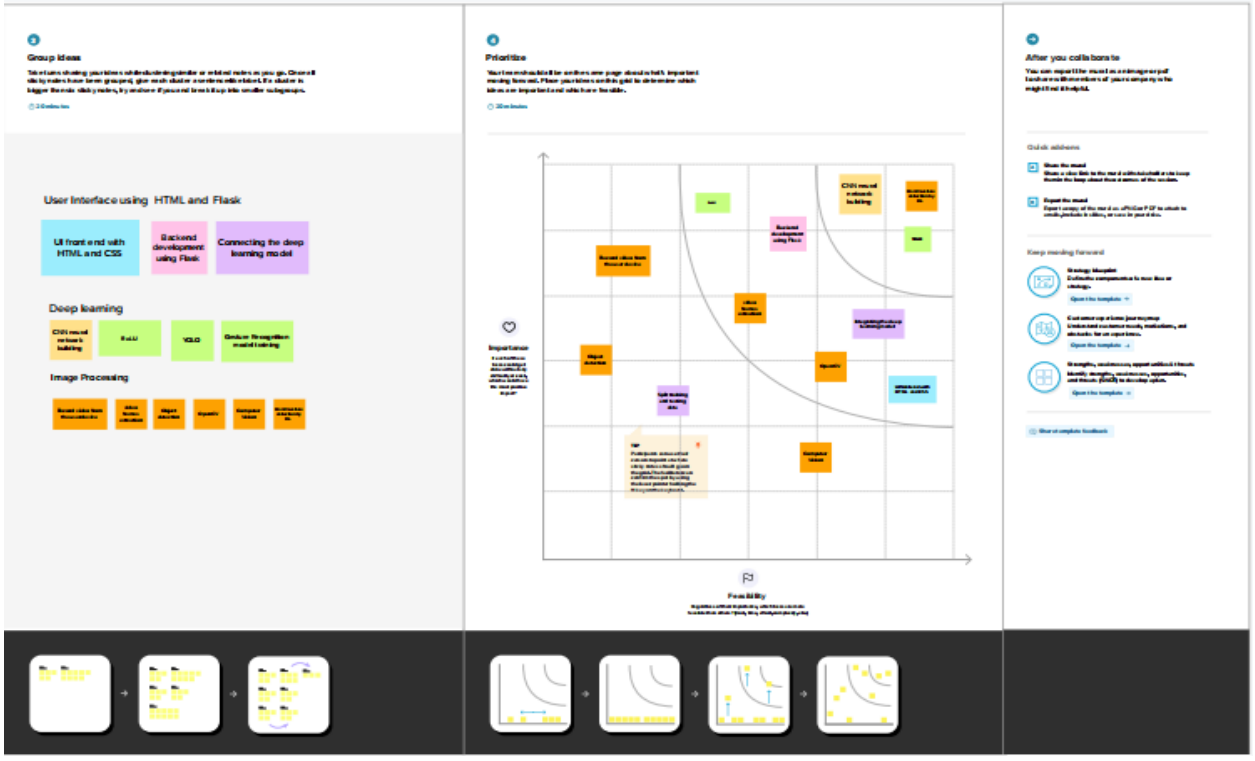
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

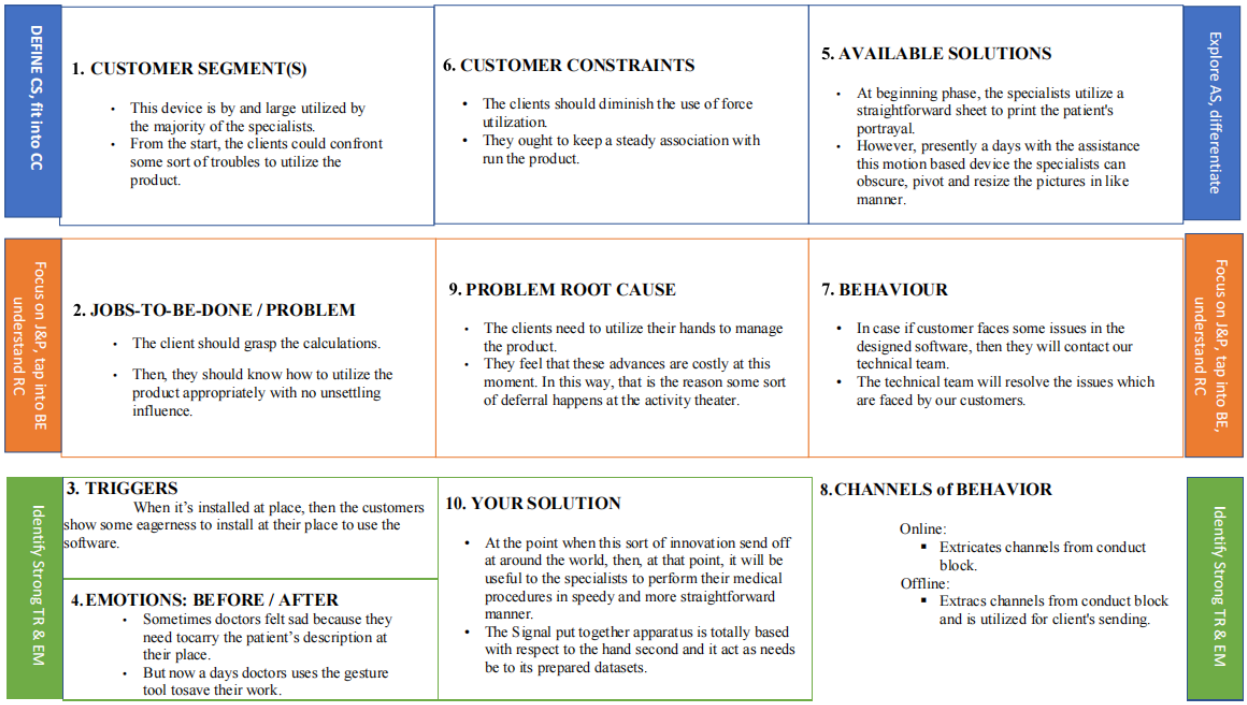




3.3 Proposed Solution

S.NO	Parameter	Description
1	Problem Statement (Problem to be solved)	Providing doctors with effective, intuitive, accurate, and safe ways of contact without compromising the quality of their job is a significant problem. However, doctors in intensive care units (ICU) frequently transfer illnesses by using computer keyboards and mice. In order to provide the highest level of sterility, the medical profession as an alternative to the current interface techniques.
2	Idea / Solution description	In order to ensure sterility, the doctor can move or manipulate the images using hand gestures.
3	Novelty / Uniqueness	Unlike other non-verbal communication methods, gestures in the operating room do not impair concentration. It works better at spotting patterns in pictures.
4	Social Impact / Customer Satisfaction	By enhancing patient care and enabling experts to regulate images without coming into direct contact with the system, it supports social responsibility by preventing the spread of diseases and hazardous radiation.
5	Business Model (Revenue Model)	Cost efficient to deploy this Software for healthcare department as well as in hospitals and can collaborate with government for health awareness camps.
6	Scalability of the Solution	Better execution in accurate results, sensitivity, system architecture design and transparency and flexibility of the software.

3.4 Problem Solution fit



4. 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	The photographs must be uploaded into the system after being captured in camera.
FR-3	Performing gestures	After categorizing, use the gesture to determine which image is accurate, and then carry out the procedure.
FR-4	Model rendering	The algorithm will begin processing the image after it has been captured.
FR-5	Sterile browsing	After recognizing the motions, the sterile browsing may be carried out.
FR-6	Visibility of images	A user can see the images after completing the process.

4.2 Non-Functional requirements

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

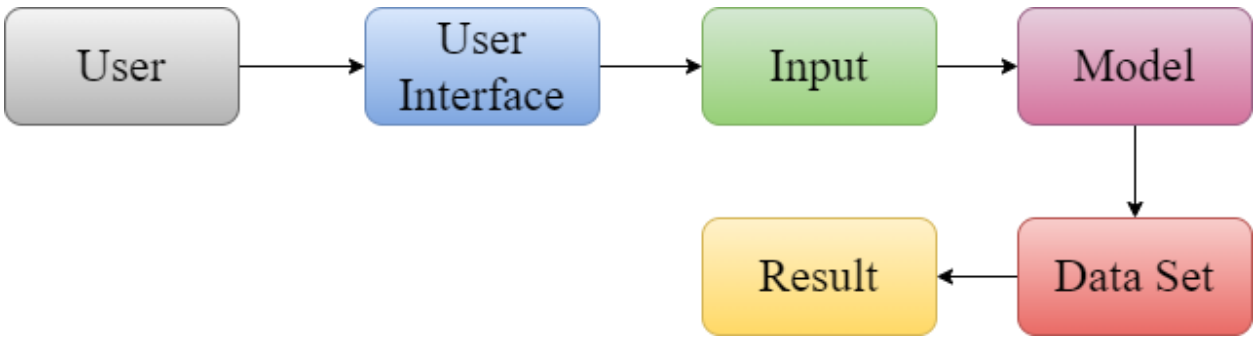
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	This system helps to control the image Prevents hazardous rays and is easy to use without direct contact with the system.
NFR-2	Security	The system is secure and can only be accessed by authorized users.
NFR-3	Reliability	Once the application is installed, the system predicts the action and provides sterile navigation.
NFR-4	Performance	The system responds to a user within seconds and the hardware and software function well.
NFR-5	Availability	An authorized user can access it anytime there is an emergency from anywhere and at any time.
NFR-6	Scalability	This technology supports a greater number of concurrent users without any discernible loss.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

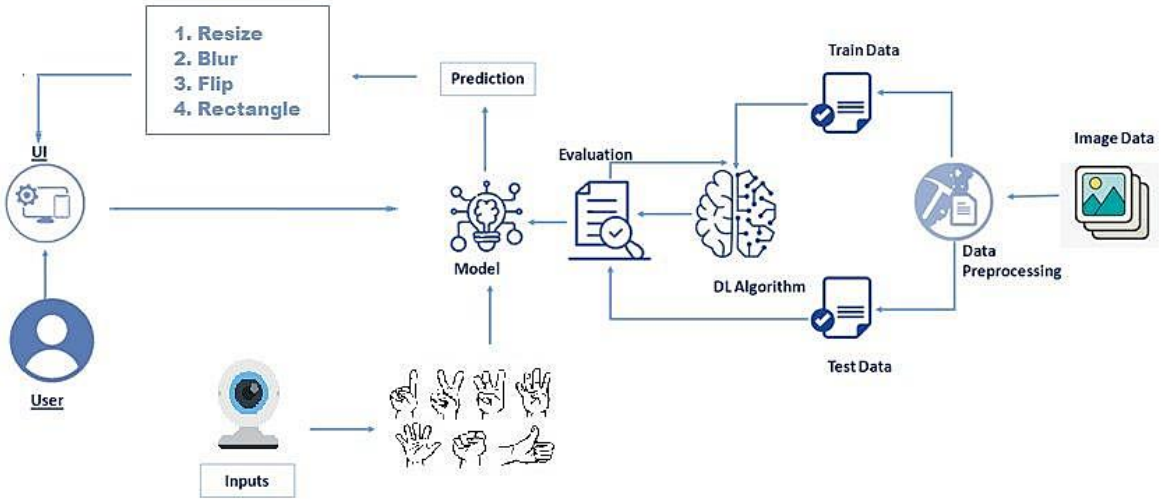
Example:Simplified



5.2 Solution & Technical Architecture

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. A two layer architecture is used: In the lower level “Gestix” provides tracking and recognition functions, while at the higher level a graphical user interface called “Gibson” manages imaging visualization.

Example - Solution Architecture Diagram



5.3 User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can upload an image for performing the action.	I can upload image	High	Sprint-3
		USN-2	As a user, I can show my hand sign in front of the camera.	I can show hand sign	High	Sprint-1 Sprint-2
		USN-3	As a user, I will send the result of the uploaded image based on my hand sign.	I can get the result	High	Sprint-2
Customer (Web user)			same as a mobile use			

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Download the Dataset	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-1		USN-2	Image Pre-processing	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-1		USN-3	Import and Configure the Image Data Generator Library and Class	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-1		USN-4	Apply Image Data Generator Functionality to Train-Set and Test-Set	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Model Building	USN-5	Import the Model Building Libraries and Initializing the Model	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-2		USN-6	Adding CNN Layers and Dense Layers	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-2		USN-7	Configure the Learning Process	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-2		USN-8	Train the Model, Save the Model and Test the Model	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-3	Application Building	USN-9	Create Web Application using HTML, CSS, JavaScript	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-3		USN-10	Build Python code	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-3		USN-11	Run the Application	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar
Sprint-4	Train The Model on IBM	USN-12	Register for IBM Cloud	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar

Sprint-4		USN-13	Train the Model and Test the Model and its Overall Performance	10	High	C.Vignesh S.Shriram U.Sanjeev G.Monish Kumar

Project Tracker ,Velocity And Burndown Chart:

Sprint	Total Story Point	Duration	Sprint Start Date	Sprint End Date	Story Point Completed	Sprint Release Date
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	5 Nov 2022	10	5 Nov 2022
Sprint-3	10	6 Days	7 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

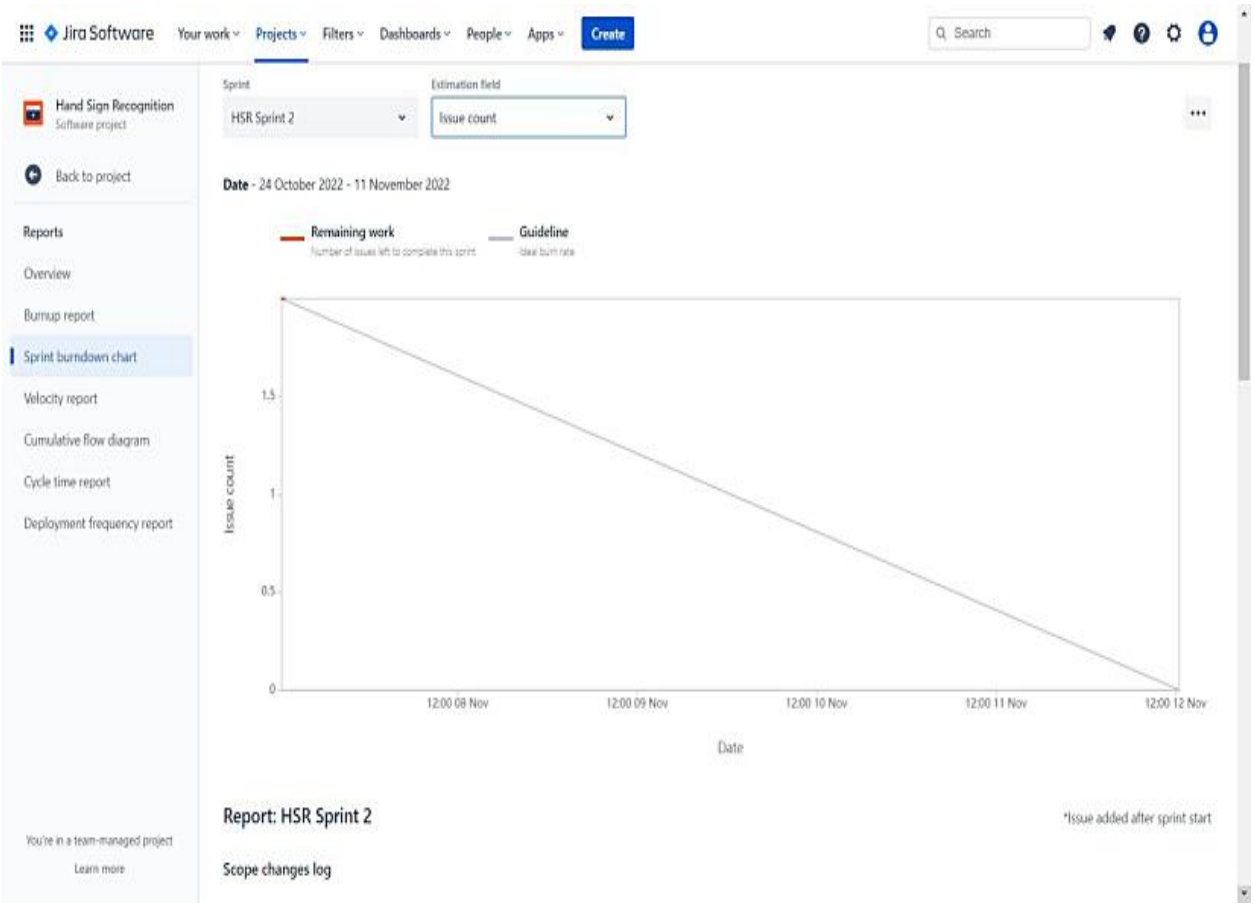
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit

(story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart



6.2 Sprint Delivery Schedule

Sprint	Total Story Point	Duration	Sprint Start Date	Sprint End Date	Story Point Completed	Sprint Release Date
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	5 Nov 2022	10	5 Nov 2022
Sprint-3	10	6 Days	7 Nov 2022	12 Nov 2022	10	12 Nov 2022

Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

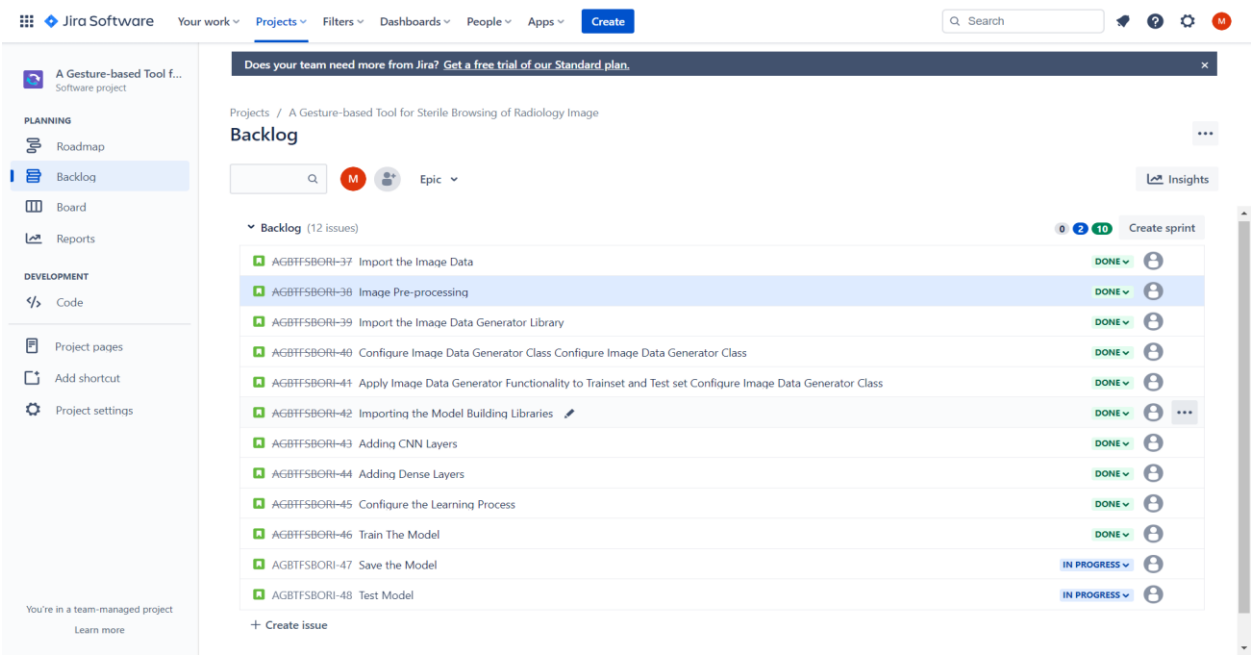
Velocity:

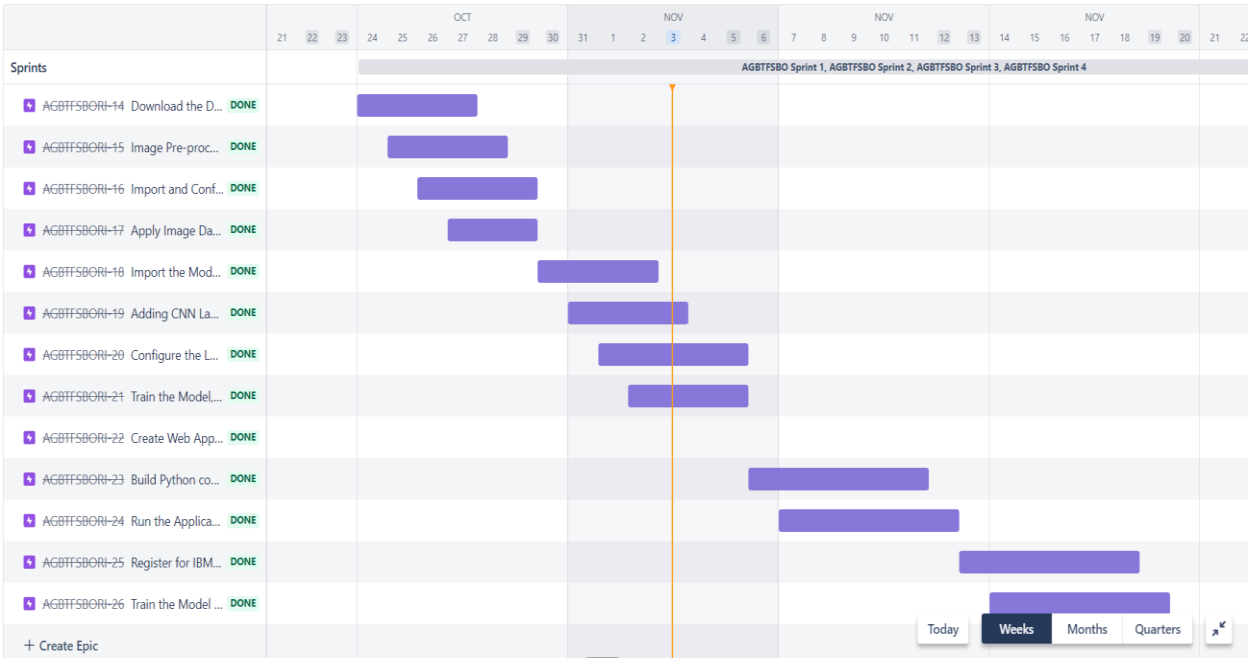
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit

(story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

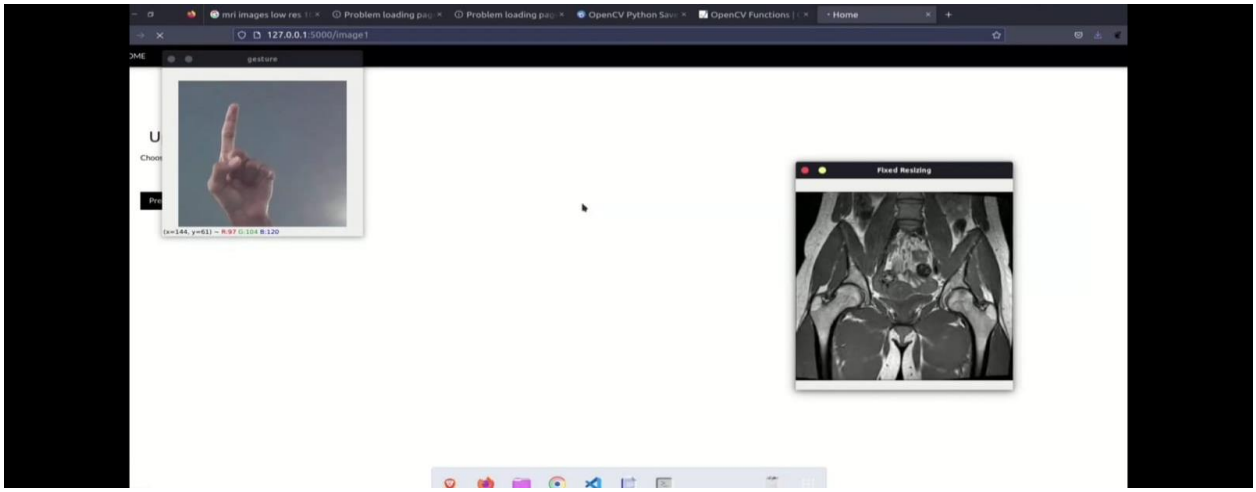
6.3 Reports from JIRA





7. CODING & SOLUTIONING

7.1 Feature 1



A gesture interface is developed for users, such as doctors/surgeons, to browse medical images in a sterile medical environment. A vision-based gesture capture system interprets user's gestures in real-time to ma-nipulate objects in an image visualization environment. A web-camera placed above a screen captures a sequence of images of the hand.

For gesture one-

The image is resized and the output is produced to the user through a window containing the manipulated image.

CODE

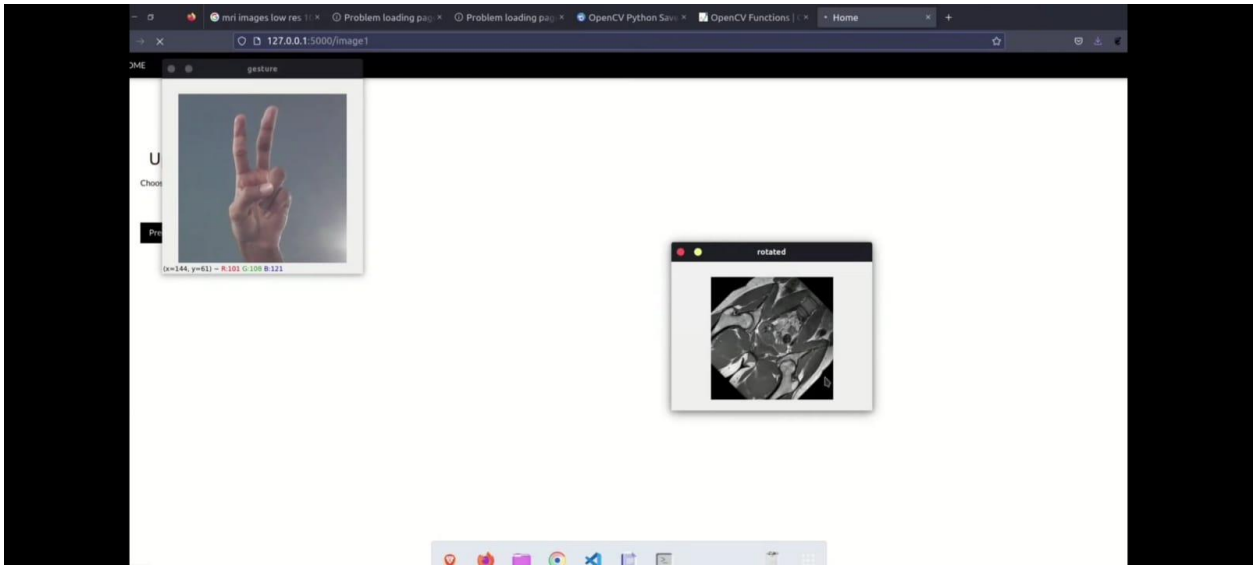
```
if res==1:
   # cv2.destroyWindow("Fixed Resizing")
   resized=cv2.resize(image1,(400,400))
   cv2.imshow("Fixed Resizing",resized)
   print("before resize")
   buffer=00

   con=False
   time.sleep(3)
   #cv2.waitKey(5000)
   # # print("after resize")
   key=cv2.waitKey(3000)
   if(key & 0xFF)==ord('1'):
      cv2.destroyWindow("Fixed Resizing")
```

7.2 Feature 2

For gesture two

The image is rotated 45degrees and the output is produced to the user through a window using openCV



CODE

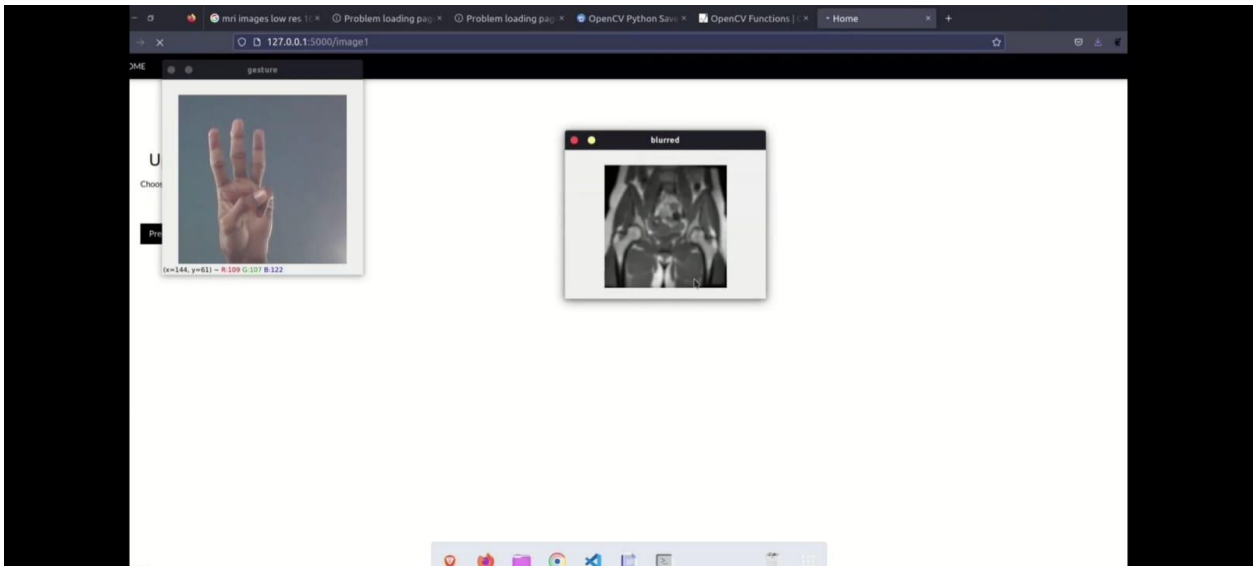
```
elif res==2:
    # cv2.destroyWindow("rotated")
    (h,w,d)=image1.shape
    center=(w//2,h//2)
    M=cv2.getRotationMatrix2D(center,-45,1.0)
    rotated=cv2.warpAffine(image1,M,(w,h))
    cv2.imshow("rotated",rotated)
    buffer=00
    # cap.release()
    con=False
    time.sleep(3)
```

7.3 Feature 3

For gesture three

The image is blurred by applying gaussianblur through openCV module of python.

The manipulated image is produced to the user via window in UI.



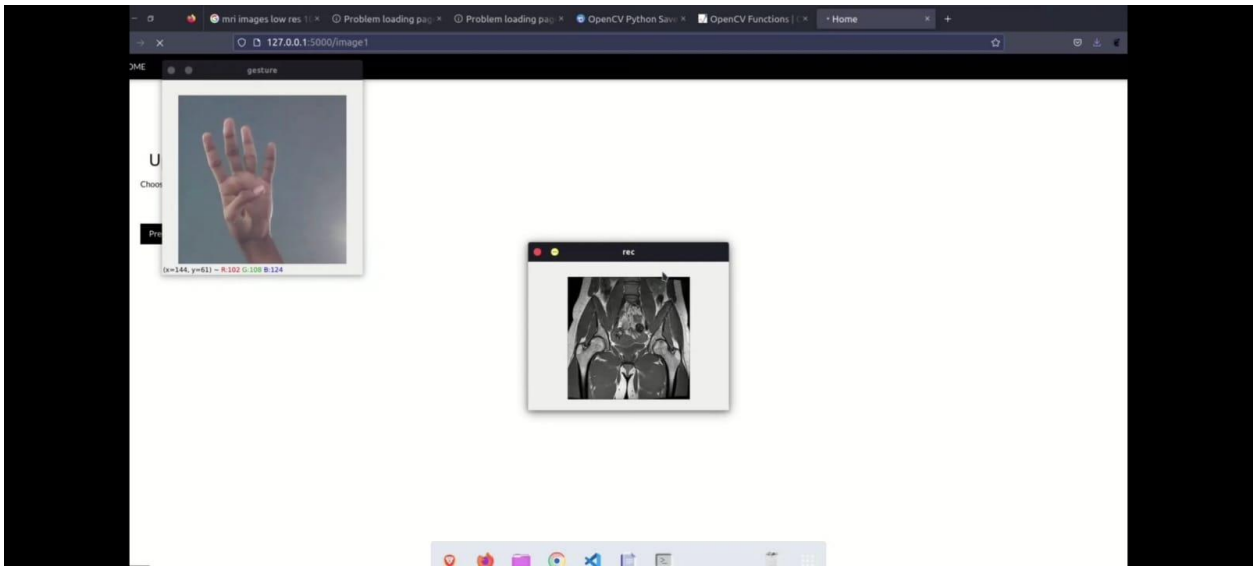
CODE

```
elif res==3:
   # cv2.destroyWindow("blurred")
   blurred=cv2.GaussianBlur(image1,(11,11),0)
   cv2.imshow("blurred",blurred)
   print("before blur")
   buffer=00
   # cap.release()
   con=False
   time.sleep(3)
```

7.4 Feature 4

For gesture four

The image is marked and an area is marked through rectangle. The box is drawn to highlight a specific area of the image. The output is produced to the user through a separate window.



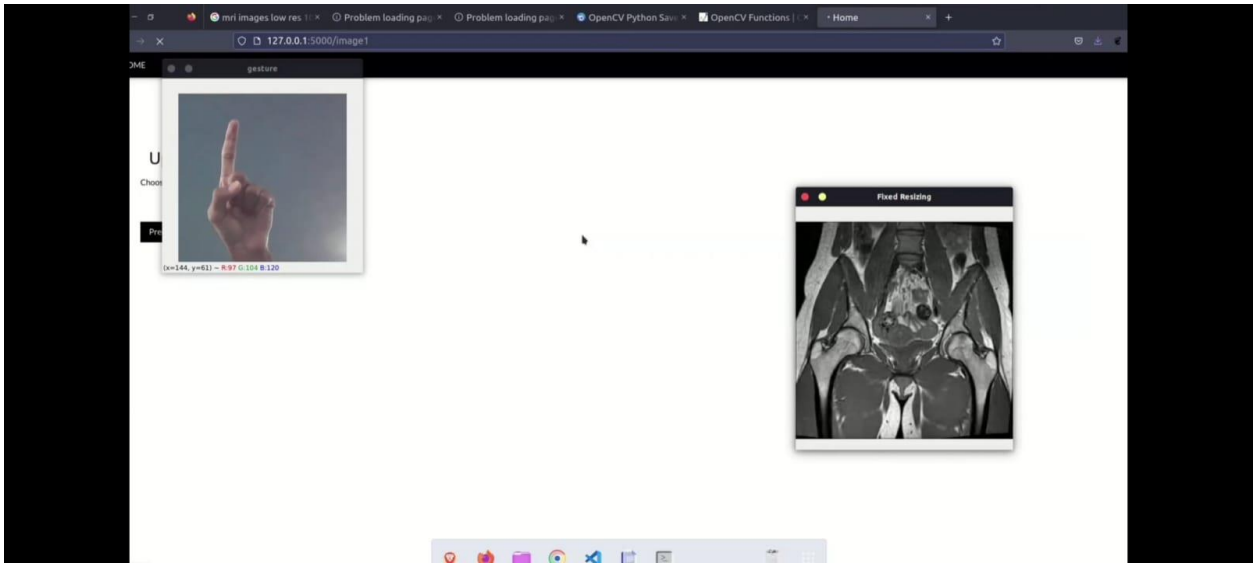
CODE

```
elif res==4:
    # cv2.destroyWindow("rectangle")
    cv2.rectangle(image1,(480,170),(650,420),(0,0,255),2)
    cv2.imshow("rectangle",image1)
    # cv2.waitKey(1)
    print("before rec")
    buffer=00
    # cap.release()
    con=False
    time.sleep(3)
```

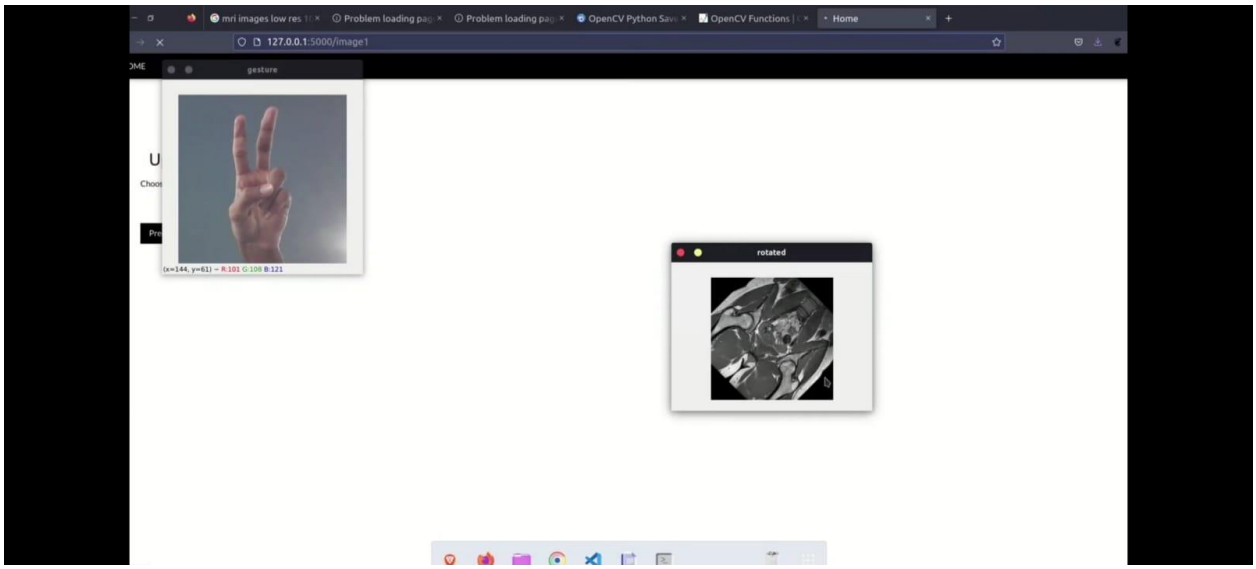
8. TESTING

8.1 Test Cases

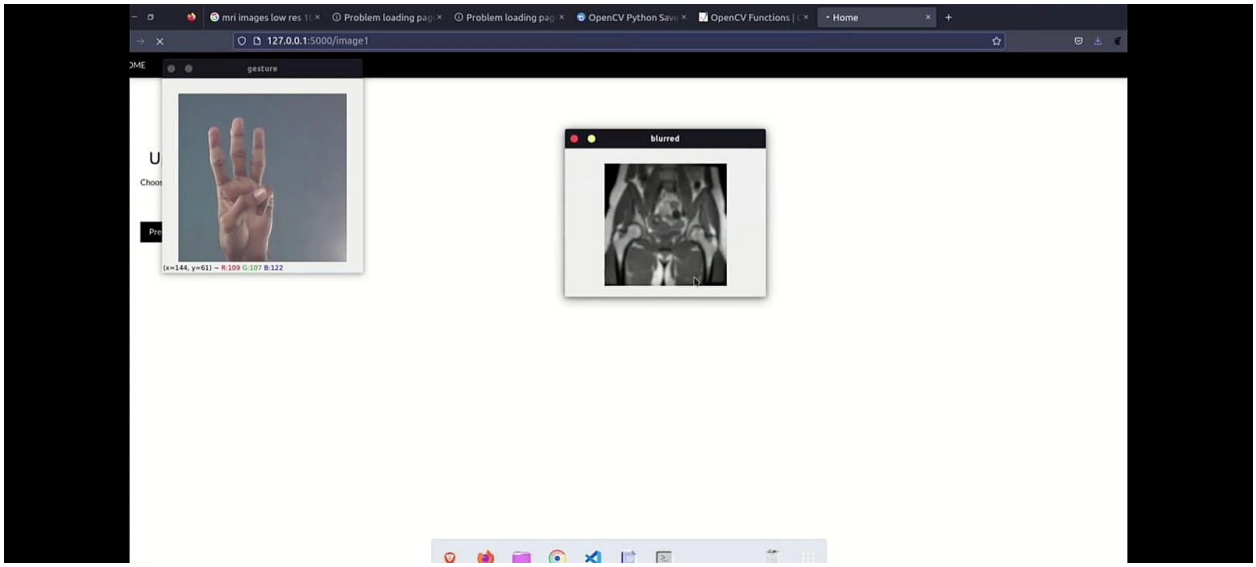
Test Case 1



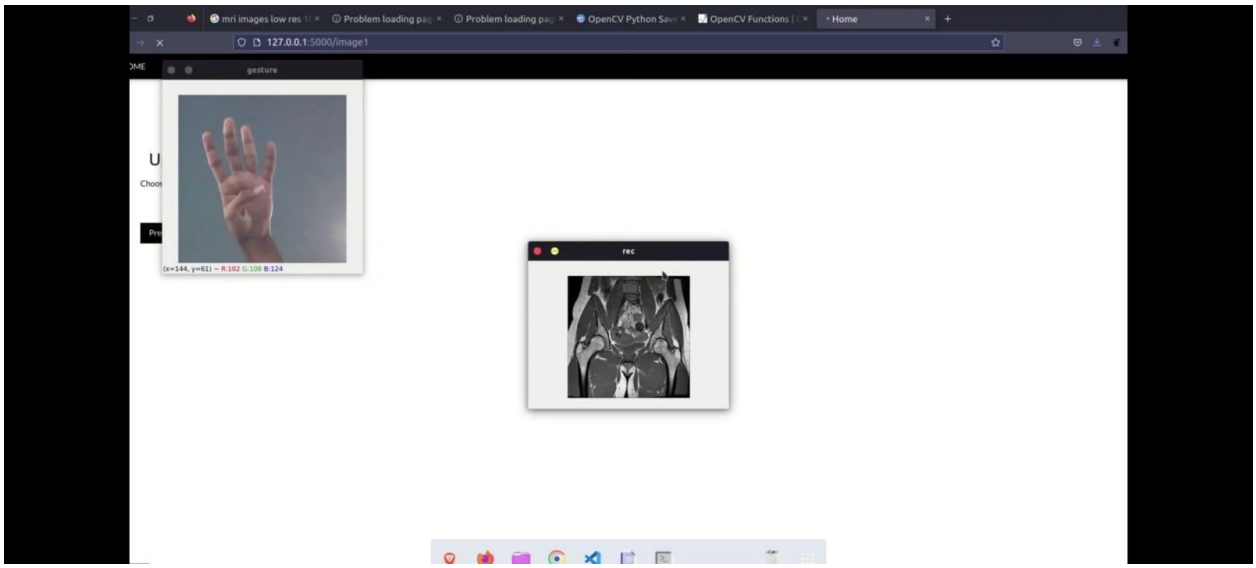
Test Case 2



Test Case 3



Test Case 4



8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] 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	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	26

3. Test Case Analysis

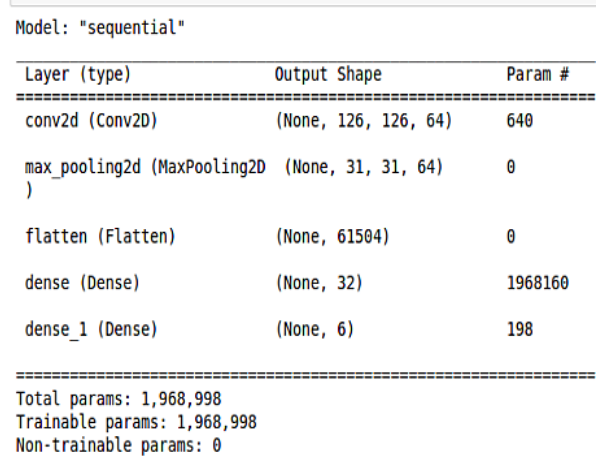
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
	Model Summary	Layer (type) Output Shape Param # ========== ====== conv2d (Conv2D) (None, 126, 126, 64) 640 max_pooling2d (MaxPooling2D (None, 31, 31, 64) 0) flatten (Flatten) (None, 61504) 0 dense (Dense) (None, 32) 1968160 dense_1 (Dense) (None, 6) 198 ========== ====== Total params: 1,968,998 Trainable params: 1,968,998 Non-trainable params: 0	

2	Accuracy	Training Accuracy - accuracy: 0.0685 precision: 0.9524 Validation Accuracy - val_accuracy: 0.2797 val_precision: 0.7267	787/787 [==========] - 263s 332ms/step - loss: 7.1602 - accuracy: 0.0189 - precision: 0. al_loss: 1.7105 - val_accuracy: 0.0036 - val_precision: 0.8583 Epoch 2/6 787/787 [==========] - 243s 309ms/step - loss: 1.0866 - accuracy: 0.0023 - precision: 0. al_loss: 1.4557 - val_accuracy: 0.0153 - val_precision: 0.8457 Epoch 3/6 787/787 [==========] - 211s 268ms/step - loss: 0.6285 - accuracy: 0.0095 - precision: 0. al_loss: 2.2450 - val_accuracy: 0.0757 - val_precision: 0.8297 Epoch 4/6 787/787 [==========] - 211s 268ms/step - loss: 0.4132 - accuracy: 0.0195 - precision: 0. al_loss: 1.6519 - val_accuracy: 0.1079 - val_precision: 0.6602 Epoch 5/6 787/787 [==========] - 186s 237ms/step - loss: 0.3160 - accuracy: 0.0377 - precision: 0. al_loss: 3.8846 - val_accuracy: 0.1733 - val_precision: 0.7175 Epoch 6/6 787/787 [==========] - 179s 227ms/step - loss: 0.2239 - accuracy: 0.0685 - precision: 0. al_loss: 8.7512 - val_accuracy: 0.2797 - val_precision: 0.7267

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

• The Hand Gesture Recognition system provides a natural way of interfacing with the computers; hence it is more User friendly.

• It provides a better computer interaction environment for the physically handicapped users.

• There is less wear and tear of the computer as the standard input devices are eliminated and a camera is used as an input device.

• Carpel-tunnel syndrome is increasing because of the repetitive use keyboard and mouse. Since this system does not use either of these as input devices it proves to be a healthier way of interacting with the computers.

• As performing gestures is human action; variations are considered till 30% tolerance. Single gesture can perform action that may require sequence of key strokes, thus helping to increase usability of system.

• Used in operations. Gurion University of the Negev (BGU) in Israel have developed a new hand gesture recognition system, tested at a Washington, D.C. hospital, that enables doctors to manipulate digital images during medical procedures by motioning instead of touching a screen, keyboard or mouse which compromises sterility and could spread infection.

DISADVANTAGES

• Such systems are difficult to develop because of the complexity and the cost of implementation.

• As each gesture is assigned a specific control command, this system is not platform independent since certain control commands vary as the operating system varies.

• Though Hand Gesture Recognition System is developed in JAVA it is not fully platform independent.

11. CONCLUSION

Gesture recognition is complicated process, as input to system cannot be determined. Using various concepts of image processing and fundamental properties of image we tried to perform this task. Success of Hand Gesture recognition system depends of end user as well as the consideration of complexity level in image processing. Currently we are living in Touch generation with implementation of Hand Gesture Recognition System for special users as well as for normal users by eliminating the traditional input devices we are going to enter into the world of Gesture Recognition Generation.

12. FUTURE SCOPE

• The system can be further enhanced to include functions with the help of which it would be possible to handle other software's (e.g. MSWORD, MSACCESS, etc) through gestures.

• We can also enhance this project such that it can translate hand gestures to Power Point commands.

• Removal of wrist band and considering another reference point and also removal of black background. • Mobile Devices, More user studies.

13. APPENDIX

Source Code

Model building and training code-

Gesture_model.ipynb

```
from keras.preprocessing.image import ImageDataGenerator

train_datagen =ImageDataGenerator(rescale=1,brightness_range=[0.4,1.5])

x_train = train_datagen.flow_from_directory(r'asldataset/training',target_size = (128,128), batch_size = 32,color_mode="grayscale", class_mode ='categorical')

test_datagen =ImageDataGenerator(rescale=1)

x_test = train_datagen.flow_from_directory(r'asldataset/testing',target_size = (128,128), batch_size = 32,color_mode="grayscale", class_mode ='categorical')

import numpy as np

import tensorflow

from tensorflow.keras.models import Sequential

from tensorflow.keras import layers

from tensorflow.keras.layers import Dense,Flatten,Conv2D,MaxPooling2D

model=Sequential()

model.add(Conv2D(64,(3,3),input_shape=(128,128,1),activation="relu"))

model.add(MaxPooling2D(pool_size=(4,4)))


model.add(Flatten())

model.add(Dense(units=32,activation="relu"))


model.add(Dense(units=6,activation="softmax"))

model.compile(optimizer="adam",loss="categorical_crossentropy",metrics=[tensorflow.keras.metrics.Accuracy(), tensorflow.keras.metrics.Precision()])

model.summary()

model.fit_generator(generator=x_train,steps_per_epoch=len(x_train),epochs=6,validation_data=x_test,validation_steps=len(x_test))

model.save("gesturemodel.h5")
```

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model=load_model("gesturemodel.h5")
img=image.load_img(r"asldataset/img.jpg",color_mode="grayscale",target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)

pred=model.predict(x)
np.argmax(pred)

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

Flask Code:

app.py

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


app=Flask(__name__,template_folder="templates")
model=load_model("temp.h5")
```

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

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

@app.route('/image1',methods=['GET','POST'])
def image1():
    return render_template("launch.html")


@app.route('/predict',methods=['GET','POST'])
def launch():
    if request.method=="POST":


        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)
        #capture video
        time.sleep(2)
        cap=cv2.VideoCapture(0)
        con=True
```

```
image1=cv2.imread(file_path)
buffer=1
zv=0
while True:
    if not con:
        cap=cv2.VideoCapture(0)
        con = True
    buffer-=1
    success,frame=cap.read()
    if success==False:
        continue
    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,(128,128))
    cv2.imshow("gesture",roi)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    # time.sleep(1)

    # success,frame=cap.read()

    cap.release()
```

```
roi=cv2.cvtColor(roi,cv2.COLOR_BGR2GRAY)
_,test_image=cv2.threshold(roi,120,255,cv2.THRESH_BINARY)
if not buffer>0:

    print("inside loop")

    result=model.predict(test_image.reshape(1,128,128,1))

    res = numpy.argmax(result)
    if numpy.amax(result) <=0.25:
        continue

    print(res)

    # if cv2.waitKey(1) & 0xFF == ord('q'):
    #     break
    if res==1:
        # cv2.destroyWindow("Fixed Resizing")
        resized=cv2.resize(image1,(400,400))
        cv2.imshow("Fixed Resizing",resized)
        print("before resize")
        buffer=00

        con=False
        time.sleep(3)
        #cv2.waitKey(5000)
        # # print("after resize")
        key=cv2.waitKey(3000)
```

```
    if(key & 0xFF)==ord('1'):
        cv2.destroyWindow("Fixed Resizing")
elif res==2:
    print("insode else",res)
    buffer=00
    con=False
    time.sleep(3)
    continue

elif res==3:
    # cv2.destroyWindow("blurred")
    blurred=cv2.GaussianBlur(image1,(11,11),0)
    cv2.imshow("blurred",blurred)
    print("before blur")
    buffer=00
    # cap.release()
    con=False
    time.sleep(3)
    #cv2.waitKey(5000)
    key=cv2.waitKey(3000)
    # print("after blur")
    if(key & 0xFF)==ord('3'):
        cv2.destroyWindow("blurred")
    # #time.sleep(5)
elif res==4:
    # cv2.destroyWindow("rectangle")
    cv2.rectangle(image1,(480,170),(650,420),(0,0,255),2)
    cv2.imshow("rectangle",image1)
```

```
    # cv2.waitKey(1)
    print("before rec")
    buffer=00
    # cap.release()
    con=False
    time.sleep(3)
    #cv2.waitKey(5000)
    key=cv2.waitKey(3000)
    print("after rec")
    if(key & 0xFF)==ord('2'):
       cv2.destroyWindow("rectangle")
    #time.sleep(5)
elif res==5:
    cv2.destroyWindow("blurred")
    cv2.destroyWindow("rotated")
    cv2.destroyWindow("rectangle")
    cv2.destroyWindow("Fixed Resizing")
    buffer=00
    # cap.release()
    con=False
    time.sleep(3)
    # break
    # time.sleep(3)
else:
    (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("rotated",rotated)
                buffer=00
                # cap.release()
                con=False
                time.sleep(3)
                #cv2.waitKey(5000)
                # print("before rotate")
                key=cv2.waitKey(3000)
                print("after rotate")
                if(key & 0xFF)==ord('2'):
                    cv2.destroyWindow("rotated")

            print("inside buffer",res)
          print("outside if buffer",res)

       cap.release()
       cv2.destroyAllWindows()
   return render_template("home.html")
if __name__ == "__main__":
  # running the app
   app.run(debug=False)
```

Front End of the website:

Home page

home.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
<title>Home</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<style>
body {font-family: "Lato", sans-serif}
.mySlides {display: none}
</style>
</head>
<body>

<!-- Navbar -->
<div class="w3-top">
  <div class="w3-bar w3-black w3-card">
    <a class="w3-bar-item w3-button w3-padding-large w3-hide-medium w3-hide-large w3-right" href="javascript:void(0)" onclick="myFunction()" title="Toggle Navigation Menu"><i class="fa fa-bars"></i></a>
    <a href="#" class="w3-bar-item w3-button w3-padding-large">HOME</a>
    <a href="{{url_for('intro')}}" class="w3-bar-item w3-button w3-padding-large w3-hide-small">TOUR</a>
    <a href="{{url_for('image1')}}" class="w3-bar-item w3-button w3-padding-large w3-hide-small">LAUNCH</a>
  </div>
</div>
```

```
<!-- Navbar on small screens (remove the onclick attribute if you want the navbar to always show on top of the content when clicking on the links) -->

<div id="navDemo" class="w3-bar-block w3-black w3-hide w3-hide-large w3-hide-medium w3-top" style="margin-top:46px">

  <a href="{{url_for('intro')}}" class="w3-bar-item w3-button w3-padding-large" onclick="myFunction()">TOUR</a>

  <a href="{{url_for('image1')}}" class="w3-bar-item w3-button w3-padding-large" onclick="myFunction()">LAUNCH</a>


</div>


<!-- Page content -->

<div class="w3-content" style="max-width:2000px;margin-top:46px">


  <!-- Automatic Slideshow Images -->

  <div class="mySlides w3-display-container w3-center">

    <img src="{{url_for('static',filename='images/img.webp')}}" style="height:800px">

    <div class="w3-display-bottommiddle w3-container w3-text-white w3-padding-32 w3-hide-small"style="max-height:1000px;">

      <h1  style="color:black;"">A Gesture-based Tool for Sterile Browsing of Radiology Images</h1>

      <h3><p style="color:blue;"><b>A web page for assisting medical professionals</b></p>   </h3>

    </div>

  </div>

  <div class="mySlides w3-display-container w3-center">

    <img src="{{url_for('static',filename='images/img1.webp')}}" style="height:800px">

    <div class="w3-display-bottommiddle w3-container w3-text-white w3-padding-32 w3-hide-small">

      <h1  style="color:black;">Gesture is recognised by deep learning.</h1>
```

```
    <h3><p style="color:blue;"><b>Real time gesture recognition is achieved by employing a well trained CNN neural network.</b></p>    </h3>

   </div>

  </div>

  <div class="mySlides w3-display-container w3-center">

   <img src="{{url_for('static',filename='images/img2.webp')}}" style="height:700px">

   <div class="w3-display-bottommiddle w3-container w3-text-white w3-padding-32 w3-hide-small">

     <h1  style="color:black;">Applications in various fields.</h1>

     <h3><p style="color:blue;"><b>The gesture recognition provides hand free image browsing facility which is highly useful for other fields.</b></p>    </h3>

   </div>

  </div>


  <!-- The Band Section -->

  <div class="w3-container w3-content w3-center w3-padding-64" style="max-width:800px" id="band">

   <h2 class="w3-wide">MOTIVATION</h2>

   <p class=" w3-justify" style="font-size: large;">Humans are able to 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.

        </p>


  </div>

<footer class="w3-container  w3-center w3-opacity w3-light-grey w3-xlarge">


  <p class="w3-content">A project done by<br><b><ul><li>Monish Kumar.G</li><li>Sanjeev .U</li><li>Shriram .S</li><li>Vignesh .C</li></ul></b></p>
```

```
</footer>

<script>
// Automatic Slideshow - change image every 4 seconds
var myIndex = 0;
carousel();

function carousel() {
  var i;
  var x = document.getElementsByClassName("mySlides");
  for (i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
  myIndex++;
  if (myIndex > x.length) {myIndex = 1}
  x[myIndex-1].style.display = "block";
  setTimeout(carousel, 4000);
}

// Used to toggle the menu on small screens when clicking on the menu button
function myFunction() {
  var x = document.getElementById("navDemo");
  if (x.className.indexOf("w3-show") == -1) {
    x.className += " w3-show";
  } else {
    x.className = x.className.replace(" w3-show", "");
  }
}
```

```
// When the user clicks anywhere outside of the modal, close it
var modal = document.getElementById('ticketModal');
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
</script>

</body>
</html>
```

Introduction page:

intro.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Home</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<style>
body {font-family: "Lato", sans-serif}
```

```
.mySlides {display: none}
</style>
</head>
<body>
  <!-- Navbar -->
<div class="w3-top">
  <div class="w3-bar w3-black w3-card">
    <a class="w3-bar-item w3-button w3-padding-large w3-hide-medium w3-hide-large w3-right" href="javascript:void(0)" onclick="myFunction()" title="Toggle Navigation Menu"><i class="fa fa-bars"></i></a>
    <a href="{{url_for('home')}}" class="w3-bar-item w3-button w3-padding-large">HOME</a>
    <a href="#" class="w3-bar-item w3-button w3-padding-large w3-hide-small">TOUR</a>
    <a href="{{url_for('image1')}}" class="w3-bar-item w3-button w3-padding-large w3-hide-small">LAUNCH</a>
  </div>
 </div>

 <!-- Navbar on small screens (remove the onclick attribute if you want the navbar to always show on top of the content when clicking on the links) -->
<div id="navDemo" class="w3-bar-block w3-black w3-hide w3-hide-large w3-hide-medium w3-top" style="margin-top:46px">
  <a href="{{url_for('home')}}" class="w3-bar-item w3-button w3-padding-large" onclick="myFunction()">HOME</a>
  <a href="{{url_for('image1')}}" class="w3-bar-item w3-button w3-padding-large" onclick="myFunction()">LAUNCH</a>

 </div>
 <!-- Page content -->
<div class="w3-content" style="max-width:2000px;margin-top:46px">
```

```
<div class="w3-black" id="tour">
  <div class="w3-container w3-content w3-padding-64" style="max-width:800px">
   <h2 class="w3-wide w3-center">About</h2>
   <p class=" w3-center"><i>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 .<br> 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.</i></p>
   <br>
  </div>
 </div>

 <h2 class="w3-wide w3-center">User Guide</h2>
 <ul class="w3-container  w3-ul w3-border w3-white w3-text-grey" >
  <li class="w3-padding"><span class="w3-tag w3-red w3-margin-left">step 1</span>Select an input image from the local system by clicking upload.</li>
  <li class="w3-padding"><span class="w3-tag w3-red w3-margin-left">step 2</span>Click the predict button. </li>
  <li class="w3-padding"><span class="w3-tag w3-red w3-margin-left">step 3</span>Once the camera window opens, show the required hand gestures. </li>
  <li class="w3-padding"><span class="w3-tag w3-red w3-margin-left">step 4</span>The desired action is performed and the resulting image is produced in a window. </li>
 </ul>
<div class="w3-container w3-content w3-padding-64" style="max-width:800px" id="contact">
 <h2 class="w3-wide w3-center">Gestures</h2>
</div>
<!-- Automatic Slideshow Images -->

 <div class="mySlides w3-display-container w3-center">
```

```
   <img src="{{url_for('static',filename='images/one.png')}}" style="height:500px">

   <div class="w3-display-bottommiddle w3-container w3-text-white w3-padding-32 w3-hide-small">

     <h1  style="color:black;">Gesture ONE.</h1>

     <h3><p style="color:blue;"><b>The provided image is resized to a fixed width and height. The resizing is done with the resize method of openCV.</b></p>    </h3>

   </div>

  </div>

  <div class="mySlides w3-display-container w3-center">

   <img src="{{url_for('static',filename='images/two.jpeg')}}" style="height:500px">

   <div class="w3-display-bottommiddle w3-container w3-text-white w3-padding-32 w3-hide-small">

     <h1  style="color:black;">Gesture TWO.</h1>

     <h3><p style="color:blue;"><b>The image is rotated by 45 degrees when the gesture two is detected.The image is rotated by getRotationMatrix2D method of openCV.</b></p>    </h3>

   </div>

  </div>

  <div class="mySlides w3-display-container w3-center">

   <img src="{{url_for('static',filename='images/three.webp')}}" style="height:500px">

   <div class="w3-display-bottommiddle w3-container w3-text-white w3-padding-32 w3-hide-small"style="max-height:1000px;">

     <h1  style="color:black;"">Gesture THREE.</h1>

     <h3><p style="color:blue;"><b>The image is blurred when the gesture 3 is detected. The image is blurred using GaussianBlur method of openCV.</b></p>
   </h3>

   </div>

  </div>

  <div class="mySlides w3-display-container w3-center">

   <img src="{{url_for('static',filename='images/four.jpg')}}" style="height:500px">
```

```
   <div class="w3-display-bottommiddle w3-container w3-text-white w3-padding-32 w3-hide-small"style="max-height:1000px;">
     <h1  style="color:black;"">Gesture FOUR.</h1>
     <h3><p style="color:blue;"><b>A rectangular area is outlined in the image. This is performed by rectangle method of openCV.</b></p>   </h3>
   </div>
  </div>
  <div class="mySlides w3-display-container w3-center">
   <img src="{{url_for('static',filename='images/five.webp')}}" style="height:500px">
   <div class="w3-display-bottommiddle w3-container w3-text-white w3-padding-32 w3-hide-small"style="max-height:1000px;">
     <h1  style="color:black;"">Gesture FIVE.</h1>
     <h3><p style="color:blue;"><b>All current windows are closed.</b></p>   </h3>
   </div>
  </div>
<br>
<div class="w3-black" id="tour">
   <div class="w3-container w3-content w3-padding-64" style="max-width:800px">
      <h2 class="w3-wide w3-center">TECHNOLOGY STACK USED:</h2>
<div class="w3-row-padding w3-padding-32" style="margin:0 -16px">
   <div class="w3-third w3-margin-bottom">
      <div class="w3-container w3-white">
        <p><b style="color:rgb(101, 101, 228);">Python</b></p>
        <p>Python is a high-level, general-purpose programming language.Python is a general-purpose high level programming language that is widely used in data science and for producing deep learning algorithms..</p>
      </div>
     </div>
     <div class="w3-third w3-margin-bottom">
```

FRONT END

The front end of the web application is designed using HTML,CSS and Javascript. The front end provides a dynamic and seamless user experience. The website contains three webpages.

openCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision. OpenCV methods are used to resize, rotate and perform other manipulations on the image based on the detected gesture.

Flask

Flask is a micro web framework written in Python. Flask is used to manage and integrate the front end with the python CNN model backend

TENSORFLOW

TensorFlow is a free and open-source software library for machine learning and artificial intelligence.It is used in developing the CNN deep learning model.

```
    <div class="w3-container w3-white">
      <p><b style="color:rgb(101, 101, 228);"> IBM cloud</b></p>
      <p>IBM cloud is used to train the deep learning model online with the dataset.</p>
    </div>
  </div>
 </div>
  </div></div>


 <script>
  // Automatic Slideshow - change image every 4 seconds
  var myIndex = 0;
  carousel();

  function carousel() {
    var i;
    var x = document.getElementsByClassName("mySlides");
    for (i = 0; i < x.length; i++) {
      x[i].style.display = "none";
    }
    myIndex++;
    if (myIndex > x.length) {myIndex = 1}
    x[myIndex-1].style.display = "block";
    setTimeout(carousel, 4000);
  }

  // Used to toggle the menu on small screens when clicking on the menu button
```

```
  function myFunction() {
    var x = document.getElementById("navDemo");
    if (x.className.indexOf("w3-show") == -1) {
      x.className += " w3-show";
    } else {
      x.className = x.className.replace(" w3-show", "");
    }
  }

  // When the user clicks anywhere outside of the modal, close it
  var modal = document.getElementById('ticketModal');
  window.onclick = function(event) {
    if (event.target == modal) {
      modal.style.display = "none";
    }
  }
  </script>
</body>
</html>
```

Prediction page:

launch.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Home</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<style>
body {font-family: "Lato", sans-serif}
.mySlides {display: none}
</style></head>
<body>
<div class="w3-top">
   <div class="w3-bar w3-black w3-card">
    <a class="w3-bar-item w3-button w3-padding-large w3-hide-medium w3-hide-large w3-right" href="javascript:void(0)" onclick="myFunction()" title="Toggle Navigation Menu"><i class="fa fa-bars"></i></a>
    <a href="{{url_for('home')}}" class="w3-bar-item w3-button w3-padding-large">HOME</a>
    <a href="{{url_for('intro')}}" class="w3-bar-item w3-button w3-padding-large w3-hide-small">TOUR</a>
    <a href="#" class="w3-bar-item w3-button w3-padding-large w3-hide-small">LAUNCH</a>
```

```
  </div>
 </div>
<div id="navDemo" class="w3-bar-block w3-black w3-hide w3-hide-large w3-hide-medium w3-top" style="margin-top:46px">
  <a href="{{url_for('home')}}" class="w3-bar-item w3-button w3-padding-large" onclick="myFunction()">HOME</a>
  <a href="{{url_for('intro')}}" class="w3-bar-item w3-button w3-padding-large" onclick="myFunction()">TOUR</a>  </div>  <br>
       <div class="w3- container" style="margin-top: 10em;margin-left: 4em;" >
        <h2 class="w3-margin-left w3-margin-top ">Upload image</h2>
      <form action = "/predict" 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"></input><br><br><br>
       <button class="w3-button w3-black w3-section w3-center" type="submit" form="upload-file" value="Submit">Predict!</button>
      </form>   </div1
       <footer> <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
 </footer>
 </html>
```

Github repository link:

https://github.com/IBM-EPBL/IBM-Project-5418-1658762482

Demo video link:

https://drive.google.com/file/d/1jGD0lL09O6bpIhKxsCMklWF7EtG1Xlr6/view?usp=drivesdk