

# **Project Report**

## **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 requirement

## **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**

## **8. TESTING**

- 8.1 Test Cases
- 8.2 User Acceptance Testing

## **9. RESULTS**

- 9.1 Performance Metrics

## **10. ADVANTAGES & DISADVANTAGES**

## **11. CONCLUSION**

## **12. FUTURE SCOPE**

## **13. APPENDIX**

Source Code  
GitHub & Project Demo Link

# **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 predictes is 1 then images is blurred;2, image is resized;3,image is rotated etc.

1. Defining our classification categories
2. Collect training images
3. Train the model
4. Test our model

## **1.2 Purpose**

The main purpose of our project is to create application in which Human can interact with system without any keyboard or mouse to achive a sterile based interaction only by using Camera, So we develop a application which only need user's hand sign to give an input throught the camera to the system.

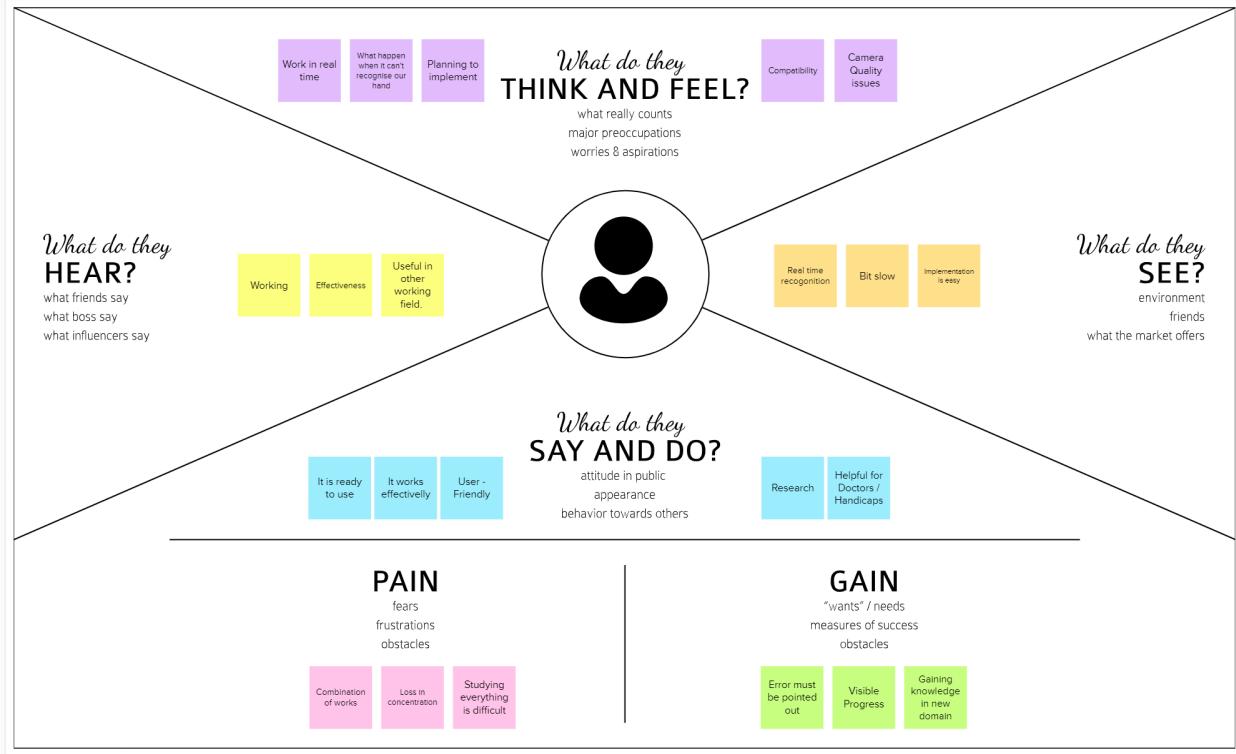
## 2. LITERATURE SURVEY

S.No	TITLE	AUTHOR(S)	TECHNIQUE(S)	YEAR	ADVANTAGE	DISADVANTAGES
1	Face Detection and recognition using OpenCV	Ramadan TH. Hasan, Amira Bibo Sallow	OpenCV, Face Detection, YOLO, Object Detection, Eigenfaces, Faster R-CNN, Fisherfaces	2021	YOLO is the most recent real-time object detection system that uses a single Neural Network to process the entire image. The SSD method is focused on the feed-forward convolutional network that generates a permanent border-box array and results in the existence of class-based entity instances in these boxes and a non-maximum deletion stage to generate final detection.	The Expected result is not always 100%.
2	Real Time Object Detection and Tracking Using Deep Learning and OpenCV	Chandan G, Ayush Jain, Harsh Jain, Mohana	Region-Based Convolution Neural Networks (RCNN), Single Shot, Detector, You Only Look Once (YOLO)	2018	Objects are detected using SSD algorithm in real time scenarios. SSD have shown results with considerable confidence level. This model showed excellent detection and tracking results on the object trained and can further utilized in specific scenarios to detect, track and respond to the particular targeted objects in the video surveillance	For making this system large amount of data is collected. And for making real time tracking high performance GPU is required.
3	Air canvas applications using OpenCV and Numpy in python	Prof.S.U.Saoji, Nishtha Dua, Akash Kumar Choudhary, Barat Phoga	Air Writing, Charater Recognition, Object Detection, Real-Time Gesture Control System, Smart Wearables, CV	2021	The system will be an excellent software for smart wearables using which people could better interact with the digital world.	Using a handwriting recognizer in place of a character recognizer will allow the user to write word by word, making writing faster. Hand-gestures with a pause can be used to control the real-time system as done by instead of using the number of fingertips. Our system sometimes recognizes fingertips in the background and changes their state

4	Two Naive Algorithm for hand tracking and gesture recognition.	Hrushikesh, Sri Vasthav Reddy, Shravalika, Sushmitha, Md Shabbeer.	OpenCV, AI, Hand Gesture Recognition	2022	Human–machine interaction (HMI) refers to the communication and interaction between a human and a machine via a user interface. This can be used to effectively and efficiently recognize and track hands and recognize hand gestures which can then be used to control computers by assigning different simple commands to different gestures.	Detecting and processing the image and also landmark the region of hand difficult to achieve
5	Hand Gesture Recognition using OpenCV and Python.	Harini V, Prahelika V, Sneha I, Adline Ebenzer P.	Histogram, Background Cancellation, Contours and Convex Hull.	2020	Histogram based approach is used to separate out the hand from the background image. Background cancellation techniques are used to produce optimum results. The detected hand is then processed and modelled by finding contours and convex hull to recognize finger and palm positions and dimensions	

### 3. IDEATION & PROPOSED SOLUTION

#### Empathy Map Canvas



#### Ideation & Brainstorming

The template consists of four main panels:

- Define Problem Statement:** A section for defining the problem with a timer (10 minutes).
- Brainstorm:** A section for generating ideas related to "GESTURE BASED DESKTOP AUTOMATION FOR BODY MOVEMENT DETECTION". It includes a grid of icons representing different gestures.
- Group Ideas:** A section for sharing ideas in a group setting. It includes a timer (20 minutes) and a note about not sharing ideas on the same page.
- Prioritize:** A section for prioritizing ideas based on importance and impact. It includes a grid for ranking ideas.

## Proposed Solution

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	Using AI, create a gesture-based desktop automation and train a model that predicts hand gestures like showing numbers with fingers as 1,2,3
2	Idea / Solution description	To develop a CNN based classifier model, which would be trained on our training data.
3	Novelty / Uniqueness	We train a CNN based model to recognize the hand gesture. The training data include images that captures the hand gestures of 1,2,3,4,5 and 0. The image is resized without much loss of information and used for training a CNN based model. We use Python Flask to provide an UI for our model.
4	Social Impact / Customer Satisfaction	This project would help the doctors in operation theatres where physical contact between persons should be avoided in order to be sterilized and also prevent from any infections.
5	Business Model (Revenue Model)	We can provide our model as a open-source or we can make this project as monthly subscription for little income.
6	Scalability of the Solution	In future, we can include face gesture detection to manipulate the image browsing.

## Problem Solution fit

Project Title: A Gesture Based tool for Sterile Browsing of Radiology Images Project Design Phase-I - Solution Fit Team ID: PNT2022TMID39288

<b>Define CS, fit into CC</b>	<b>1. CUSTOMER SEGMENT(S)</b> Our customers are Doctors especially Surgeons.	<b>6. CUSTOMER CONSTRAINTS</b> To use gestures in the right context, customers must remember many gestures. The camera is needed to accurately capture the gestures.	<b>5. AVAILABLE SOLUTIONS</b> Doctors can use the device, but doing so could infect them. They need to use a different device or ask another person to change the pictures while he is performing procedure.	<b>Explore AS, differentiate</b>
<b>Focus on J&amp;P, tap into BE, understand RC</b>	<b>2. JOBS TO BE DONE / PROBLEMS</b> In order to avoid customers from getting into contact with infection, the system enables the users to gesture based on the tools that are selected while browsing radio logical images.	<b>9. PROBLEM ROOT CAUSE</b> The real reason the problem exists because of the problems in the doctor there are no many number of technology experts in their domain.	<b>7. BEHAVIOUR</b> Customers are given a well equipped guideline to help them with their questions and concerns. we also give them a necessary training for how to work with the app.	<b>Focus on J&amp;P, tap into BE, understand RC</b>
<b>Identity story TR&amp;EM</b>	<b>3. TRIGGERS</b> The time-efficient and easy browsing trigger the customers to switch to this technology.	<b>10. YOUR SOLUTION</b> If you are working on an extended business, its still less cost so we can think the solution is to design a gesture based tool for browsing. The surgeon only has to show his hand sign and the machine will detect the hand sign and change the images accordingly.	<b>8. CHANNELS OF BEHAVIOUR</b> 8.1 ONLINE A wireless connection is required to analyze and select the tool from the compendium of radio images.  8.2 OFFLINE Doctors need to use their old prefer way.	<b>Identity story TR&amp;EM</b>
<b>Focus on J&amp;P, tap into BE, understand RC</b>	<b>4. EMOTIONS: BEFORE / AFTER</b>  BEFORE: frustration, Angry, worried about the patient.  AFTER: Feeling satisfaction, Getting full concentration on the job.			

## **4. REQUIREMENT ANALYSIS**

### **Functional requirements**

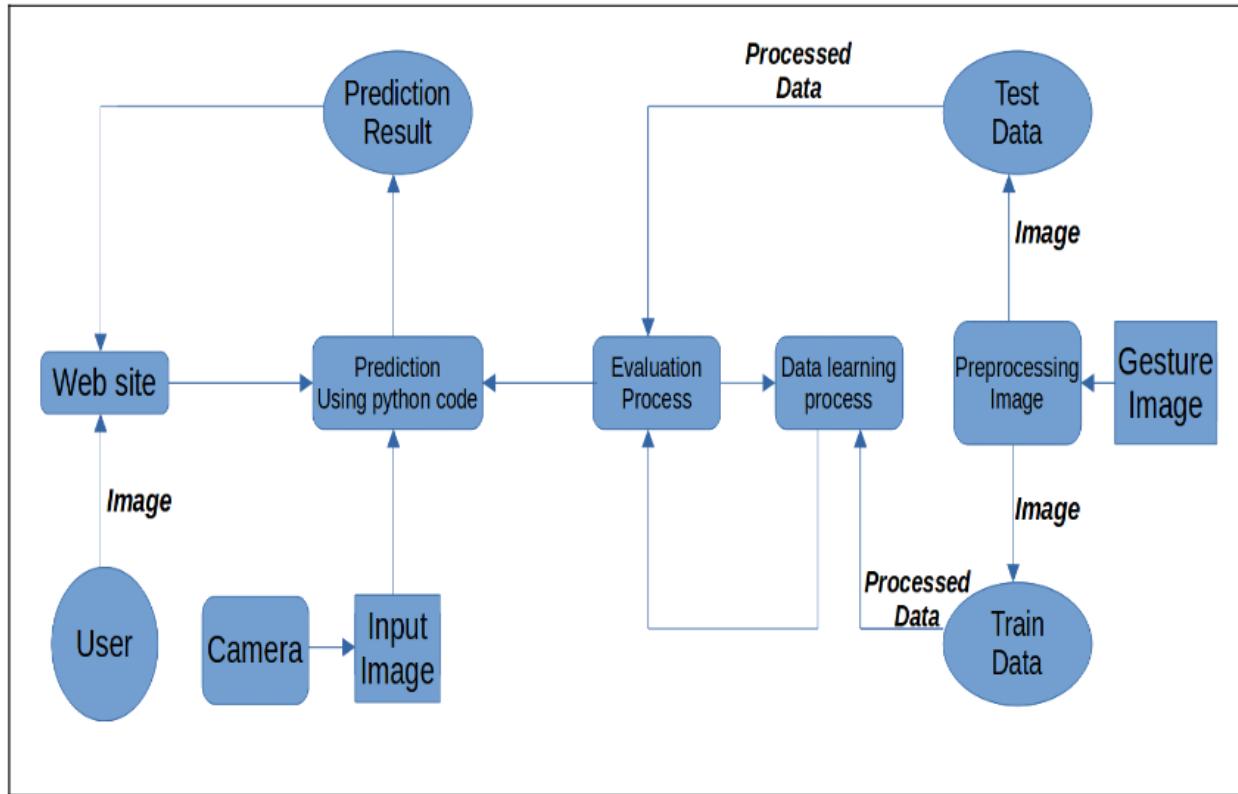
<b>S.no</b>	<b>Parameters</b>	<b>Functional Requirement</b>
1.	User Requirements	pc, camera for input
2.	Hardware requirements	min: i3 4 <sup>th</sup> gen processor, 2Gb RAM
3.	Software requirements	python3, browser
4.	Module requirements	tensorflow, matplot, opencv, os, flask

### **Non-Functional requirements**

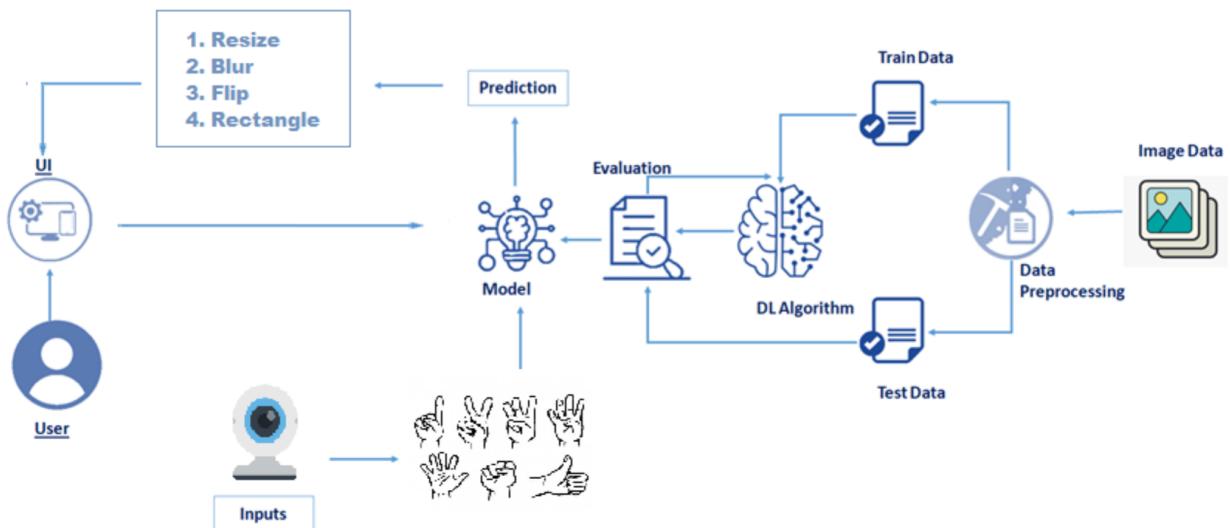
<b>Steps</b>	<b>Non-Functional Requirement</b>
1	Run the python application
2	open the browser
3	go to localhost:5000 to access user interface
4	go to launch page and upload the image
5	click on predict.

# 5. PROJECT DESIGN

## Data Flow Diagrams



## Solution & Technical Architecture



## User Stories

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Tas	Story Points	Priority	Team Members
Sprint-1	User Interface Building	USN-1	As a user, I can use this website easily	10	Medium	Ganesh J Abinash M Anandaraj E Krishnakumar G
Sprint-3	Additional function	USN-12	As a user, I can use the website to blur the image and its really amazing	10	Low	Ganesh J Abinash M Anandaraj E Krishnakumar G
Sprint-2	Deployment of AI model in the cloud	USN-3	As a user i love to use a web app for human computer interaction	20	High	Ganesh J Abinash M Anandaraj E Krishnakumar G
Sprint-1	User Interface Building	USN-4	I think need better user interface for mobile	10	Low	Ganesh J Abinash M Anandaraj E Krishnakumar G
Sprint-4	Additional function	USN-5	Need some additional function like crop	10	Low	Ganesh J Abinash M Anandaraj E Krishnakumar G
Sprint-3	Deployment of AI model in the cloud	USN-6	As a user, I think the website need to faster in back end processing	20	High	Ganesh J Abinash M Anandaraj E Krishnakumar G
Sprint-4	Prediction	USN-7	As a user, I can get the predicted results from the model deployed in the cloud	20	High	Ganesh J Abinash M Anandaraj E Krishnakumar G

## 6. PROJECT PLANNING & SCHEDULING

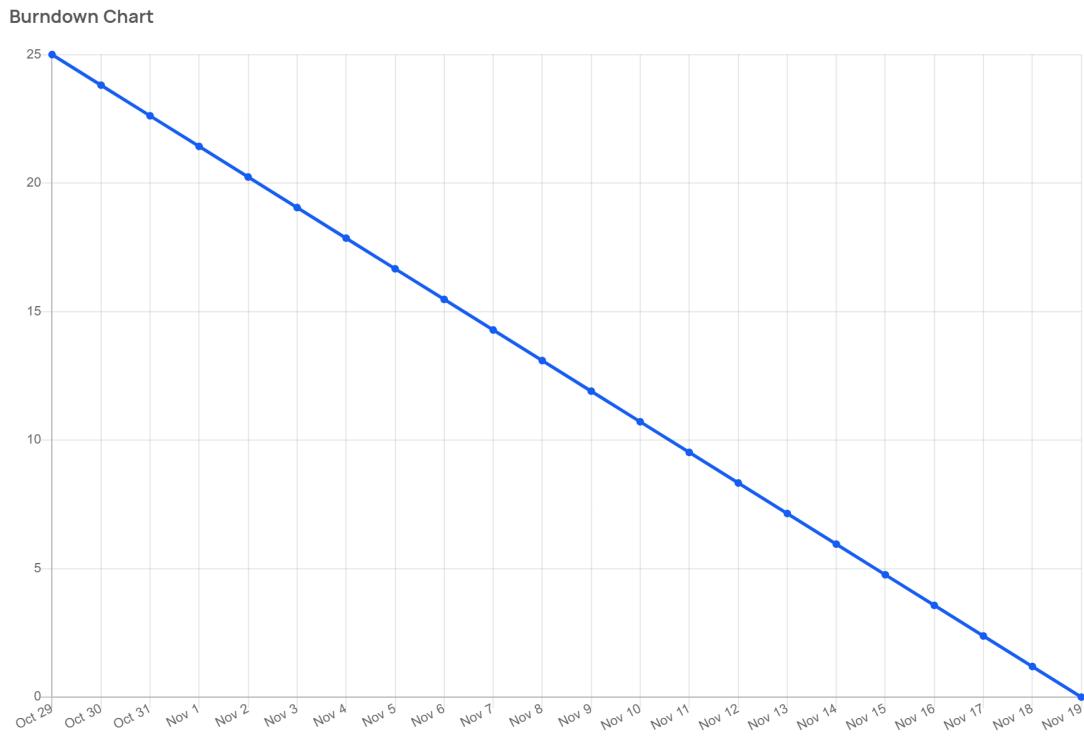
### Sprint Planning & Estimation

S.no	Milestone	Activites	Date
1.	Preparation Phase	Pre-requisites	22 Aug 2022 - 27 Aug 2022
		Prior Knowledge	
		Project Structure	
		Project Flow	
		Project Objective	
		Registrations	
		Environment Setup	
2.	Ideation Phase	Literature Survey	29 Aug 2022 - 03 Sep 2022
		Empty Map	05 Sept 2022 - 10 Sept 2022
		Ideation	12 Sept 2022 - 17 Sept 2022
3.	Project Design Phase - I	Proposed Solution	19 Sept 2022 - 24 Sept 2022
		Problem Solution Fit	19 Sept 2022
		Solution Architecture	
4.	Project Design Phase - II	Customer Journey	03 Oct 2022 - 08 Oct 2022
		Requirement Analysis	10 Oct 2022 - 15 Oct 2022
		Data Flow Diagrams	
		Technology Architecture	
5.	Project Planning Phase	Milestone & Tasks	17 Oct 2022 - 22 Oct 2022
		Sprint Schedules	
6.	Project Development Phase	Sprint - 1	24 Oct 2022 - 29 Oct 2022
		Sprint - 2	31 Oct 2022 - 05 Nov 2022
		Sprint - 3	07 Nov 2022 - 12 Nov 2022
		Sprint - 4	14 Nov 2022 - 19 Nov 2022

### Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint - 1	20	6 Days	24 Oct 2022	29 Oct 2022	20	27 Oct 2022
Sprint - 2	20	6 Days	31 Oct 2022	05 Nov 2022	20	04 Oct 2022
Sprint - 3	30	6 Days	07 Nov 2022	12 Nov 2022	30	09 Oct 2022
Sprint - 4	30	6 Days	14 Oct 2022	19 Nov 2022	30	14 Oct 2022

## Reports from JIRA



Burndown Chart

## 7. CODING & SOLUTIONING

### Required Modules :

```
from flask import Flask,render_template,request
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
import os
from werkzeug.utils import secure_filename
```

These are the important modules we required to run the application.

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

### Initialize the Flask :

It will initialize the flask web server frame work for the user interface

```
@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")
```

### **Request and Response function :**

These are function which helps to receive the request from the browser and send the appropriate response to the browser.

### **Prediction model :**

```
# Loading the model  
model=load_model('gesture.h5')
```

This Statement will Include the Trained model.

### **Camera access :**

```
cap = cv2.VideoCapture(0)  
while True:  
    _, frame = cap.read() #capturing the video frame values  
    frame = cv2.flip(frame, 1)# Simulating mirror image  
    x1 = int(0.5*frame.shape[1]) # Coordinates of the ROI  
    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)# Drawing the ROI  
    roi = frame[y1:y2, x1:x2]# Extracting the ROI
```

This code helps to access the camera to get an input for the prediction model.

### **Upload the image :**

```
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)
```

Its work is to save the uploaded image for the manipulation.

### Prediction function :

```
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)
```

It will predict the value of the input from the camera

### Manipulation code :

```
if prediction[0][0]=='ZERO':
    cv2.waitKey(0)
```

If the prediction value is zero it stop the process and wait for the key 0 to press.

```
elif prediction[0][0]=='ONE':
    try:
        cv2.destroyAllWindows("op")
    except:
        print("1")
    finally:
        resized = cv2.resize(image1, (200, 200))
        cv2.imshow("op", resized)
elif prediction[0][0]=='THREE':
    try:
        cv2.destroyAllWindows("op")
    except:
        print("3")
    finally:
        (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("op", rotated)
```

- If the prediction value is one then the image will be resized to shape 200X200.
- If the prediction value is three then the image will be rotated to angle 45degree.
- If the prediction value is two then the image will be blurred.
- If the prediction value is four then the image will be resized to 400X400.

```

elif prediction[0][0]=='FIVE':
    try:
        cv2.destroyAllWindows("op")
    except:
        print("5")
    finally:
        gray = cv2.cvtColor(image1, cv2.COLOR_RGB2GRAY)
        cv2.imshow("op", gray)

else:
    continue

```

- If the prediction value is five then the image will be converted to rgb to gray.
- If the prediction value is something else it just continue.

## User Interface :

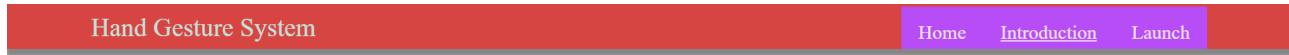
### Home Page :



**HAND GESTURE RECOGNITION  
OF  
RADIOLOGY IMAGES  
THROUGH  
STERILE BROWSING**

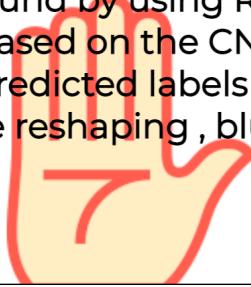
---

## **Introduction Page :**



### **INTRODUCTION**

Hand Gesture recognition system provides us an innovative, natural, user friendly way of interaction with the computer which is more familiar to the human beings. In our project, the hand region is extracted from the background by using Region of interest. 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.



## **Launch Page**



### **Hand Gesture Recognition**

Provide an image for which you want to perform various operations

**Upload Image Here**

Choose...

Upload Image Here

Choose...



Predict!

In this page you can upload the image to be edited, after upload the image just press predict.

## **8. TESTING**

### **Test Cases :**

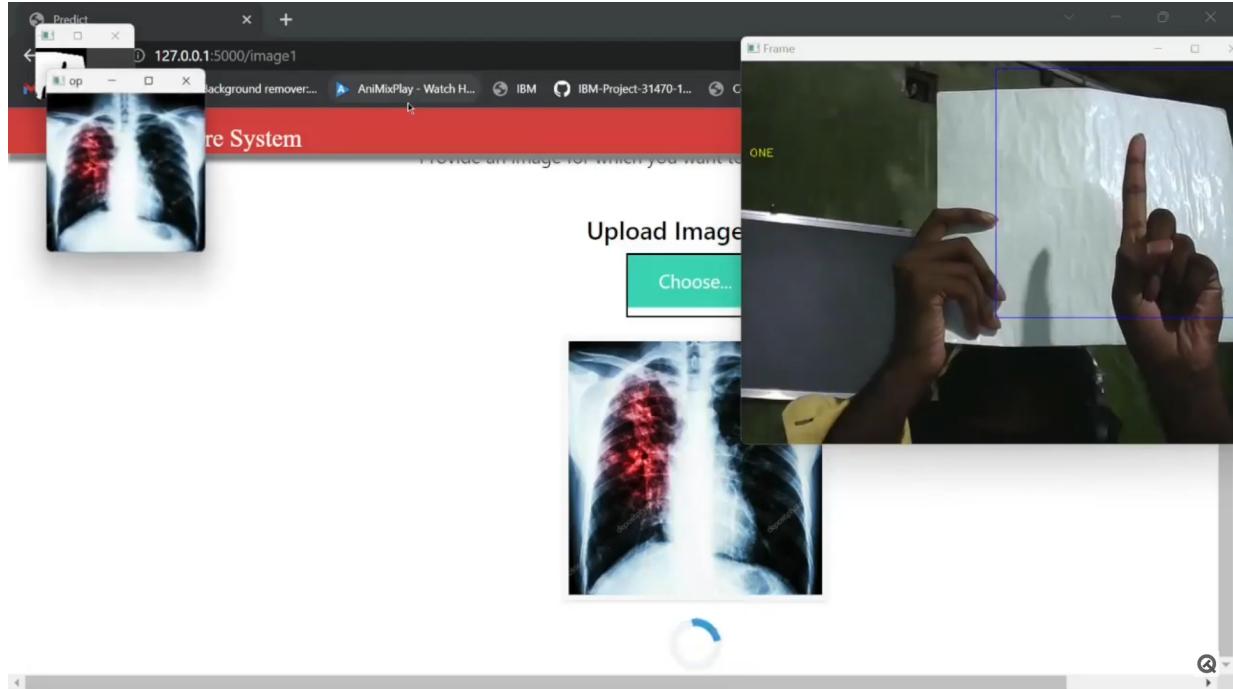
Refer : Testcases Report.xlsx

### **User Acceptance Testing :**

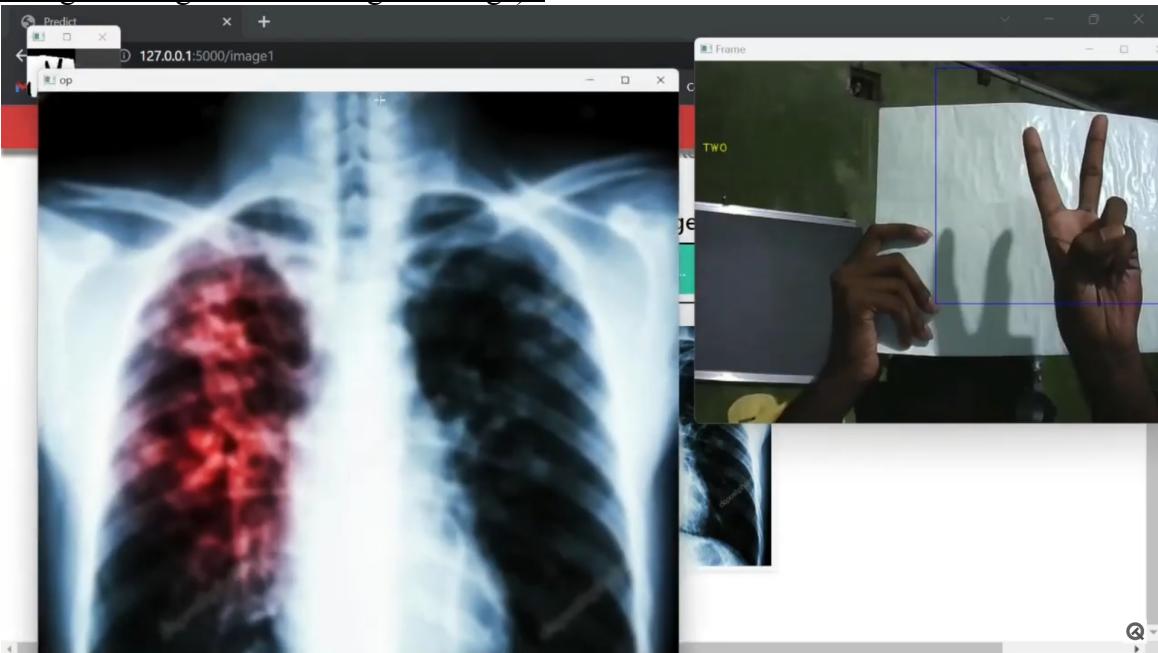
Refer : UAT Report.pdf

## **9. RESULTS**

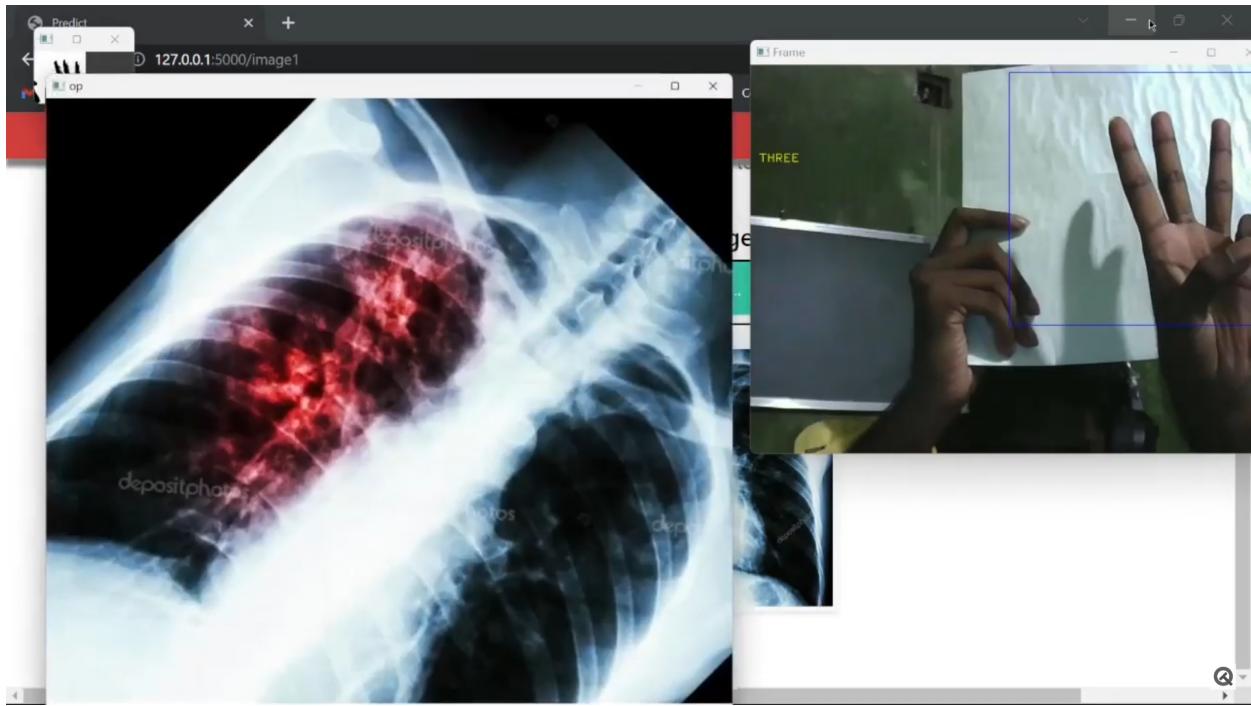
1. (Using One finger -- Opening Image):



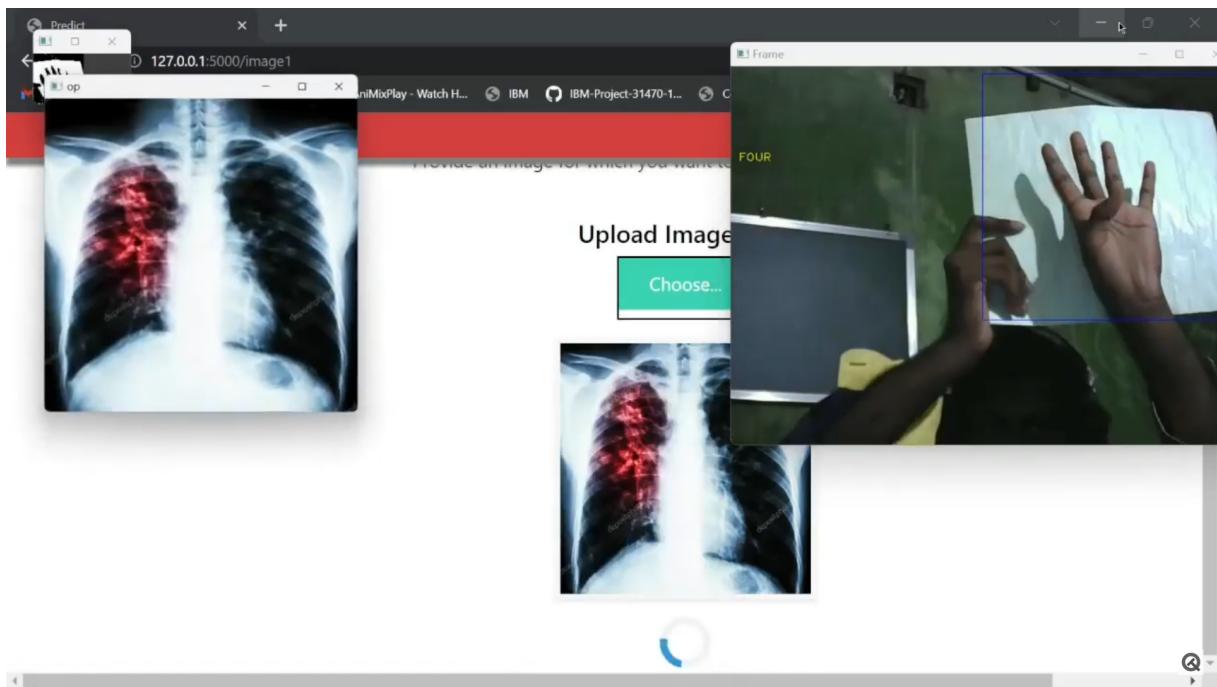
2. (Using two fingers -- Zooming the image) :



3. (Using 3 Fingers -- Rotating The Image) :



#### 4. (Using four Fingers) :



## **Performance Metrics :**

Model Performance Testing :

S.no	Parameter	Values	Screenshots
1.	Model Summary	Total Params : 3,224,422 Trainable Params : 3,224,422	Fig - 9.1
2.	Accuracy	Training Accuracy - 98%	Fig - 9.2 (a)
		Validation Accuracy - 96%	Fig - 9.2 (b)

### **Summary**

```
In [8]: model.summary()
Model: "sequential"
Layer (type)          Output Shape         Param #
=====
conv2d (Conv2D)        (None, 62, 62, 32)      320
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)    0
)
conv2d_1 (Conv2D)       (None, 29, 29, 32)     9248
max_pooling2d_1 (MaxPooling2D) (None, 14, 14, 32)    0
)
flatten (Flatten)       (None, 6272)           0
dense (Dense)          (None, 512)            3211776
dense_1 (Dense)         (None, 6)              3078
=====
Total params: 3,224,422
Trainable params: 3,224,422
Non-trainable params: 0
```

Fig - 9.1

```
In [10]: train_loss = his.history['loss']
val_loss = his.history['val_loss']
train_acc = his.history['accuracy']
val_acc = his.history['val_accuracy']
xc = range(1,26)

plt.figure()
plt.plot(xc, train_loss)
plt.plot(xc, val_loss)

Out[10]: [
```

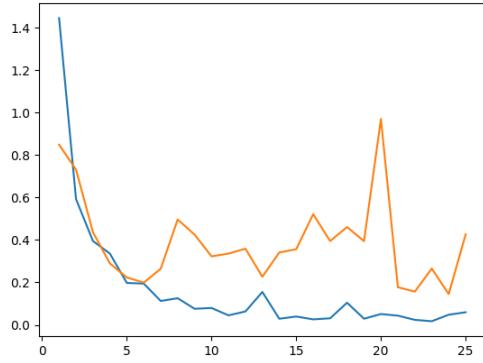


Fig - 9.2 (a)

```
In [11]: plt.plot(xc, train_acc)
plt.plot(xc, val_acc)

Out[11]: [
```

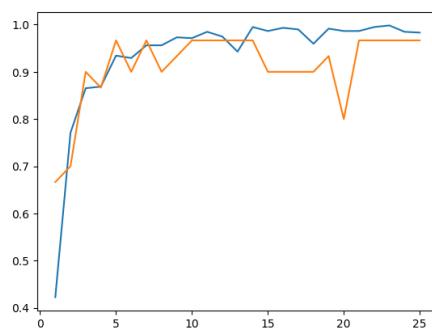


Fig - 9.2 (b)

## **10. ADVANTAGES & DISADVANTAGES**

### **Advantages :**

- As mention in Problem Statement, The Images can be viewed, Zoom, operated Without using Keyboard or Mouse for acheiving Sterile Browsing.
- Not only Radiology Images, But also Other Image Files can be uploaded and Viewed.
- Can be easily implemented on low powered processor.
- Works in Both Linux and Windows.

### **Disadvantages :**

- Light position should be correctly positioned in order to improve accuracy.
- Sometimes it will stop automatically after certain period of time on running, In order to Continue, there should be some Input in command prompt (Terminal, PS).

## **11. CONCLUSION**

So the main objective of our Project is to create a application in which the user uses their hands to give input to the system to browse Images. This is performed in order to achieve Sterile Browsing. Hence The application is built in Industrial method by following Agile methodology to understand the basic concepts of Agile methodology, and programming concepts of Python, Tensorflow . And also train and implement the model on IBM Cloud etc.

## **12. FUTURE SCOPE**

- In Future, Certain Disadvantages Can be removed, and achieve full Sterile Browsing.
- Might be implemented not only for sterile browsing but also for other applications.
- Can be implemented on AR/VR 3D Tracking Technologies

## **13. APPENDIX**

### **Source Code**

**GitHub Link :** <https://github.com/IBM-EPBL/IBM-Project-31470-1660200782>

**Project Demo :** <https://youtu.be/2syVe3xQYWE>