

# **A GESTURE- BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES**

**(TEAM ID : PNT2022TMID52914)**

## **PROJECT REPORT**

**Submitted by**

<b>Bharath Srivathsan D</b>	<b>193002016</b>
<b>Ela Ezhilarasi</b>	<b>193002025</b>
<b>G Isha Ambika</b>	<b>193002037</b>
<b>A Jeyadevan</b>	<b>193002041</b>

*in partial fulfillment for the award of degree*

*of*

**BACHELOR OF ENGINEERING**

**In**

**Electronics and Communication Engineering**



**SRI SIVASUBRAMANIA NADAR COLLEGE OF ENGINEERING, CHENNAI**

*Affiliated to Anna University, Chennai*

# INDEX

## **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**

## **8. TESTING**

## **9. RESULTS**

## **10. ADVANTAGES & DISADVANTAGES**

## **11. CONCLUSION**

## **12. FUTURE SCOPE**

## **13. APPENDIX**

Source Code

GitHub & Project Demo Link

# **CHAPTER 1**

## **INTRODUCTION**

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. In this project, 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.

In this project, we refer to gestures as a basic form of non-verbal communication made with the hands. Psychological studies showed that young children use gestures to communicate before they learn to talk. Manipulation, as a form of gesticulation, is often used when people speak to each other about some object. Naturalness of expression, non-encumbered interaction, intuitiveness and high sterility are all good reasons to replace the current interface technology (e.g., keyboard, mouse, and joystick) with more natural interfaces.

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

### **1.2 Purpose**

Purpose of the project is used to browse the radiology images using hand gestures rather than using mouse, keyboard etc, to maintaining sterility inside the operation room.

## CHAPTER 2

### LITERATURE SURVEY

1. Systematic analysis on 55 scientific papers and 33 journal publications that focus on touchless human–computer interaction in operating rooms and interventional radiology suites was performed. Most of the identified literature (62 %) deals with the control of medical image viewers. The others present interaction techniques for laparoscopic assistance, telerobotic assistance and operating room control as well as for robotic operating room assistance and intraoperative registration.
2. This paper proposes a new medical image control concept based on a Brain Computer Interface (BCI) that allows for hands-free and direct image manipulation without relying on gesture recognition methods or voice commands. This yielded promising results, and showed its potential for future intraoperative applications.
3. In this work, three different interaction modes for image manipulation that are usable in a surgery setting:
  1. gesture-controlled approach using Kinect ;
  2. oral instructions to a third part dedicated to manipulate the images; and
  3. direct manipulation using a mouse.

The study shows with formal evaluation that the use of gestures is advantageous over instructions to a third person. The given gestures could be learned easily and reliability of the tested gesture-control system is good.

## **2.1 Existing problem**

When the doctors inside the operation room use keyboard and mouse, there is a possibility of spreading infection. So, in this project we used hand gesture methodology to keep the doctor sterile.

## 2.2 References

1. Mewes, A., Hensen, B., Wacker, F., & Hansen, C. (2016). *Touchless interaction with software in interventional radiology and surgery: a systematic literature review*. *International Journal of Computer Assisted Radiology and Surgery*, 12(2), 291–305.
2. *Introducing a brain-computer interface to facilitate intraoperative medical imaging control – a feasibility study* Hooman Esfandiari, Pascal Troxler, Sandro Hodel, Daniel Suter, Mazda Farshad, Collaboration Group and Philipp Fürnstahl *BMC Musculoskeletal Disorders*, Volume 23, Article Number:701(2022)
3. *Gesture-Controlled Image Management for Operating Room: A Randomized Crossover Study to Compare Interaction Using Gestures, Mouse, and Third Person Relaying*

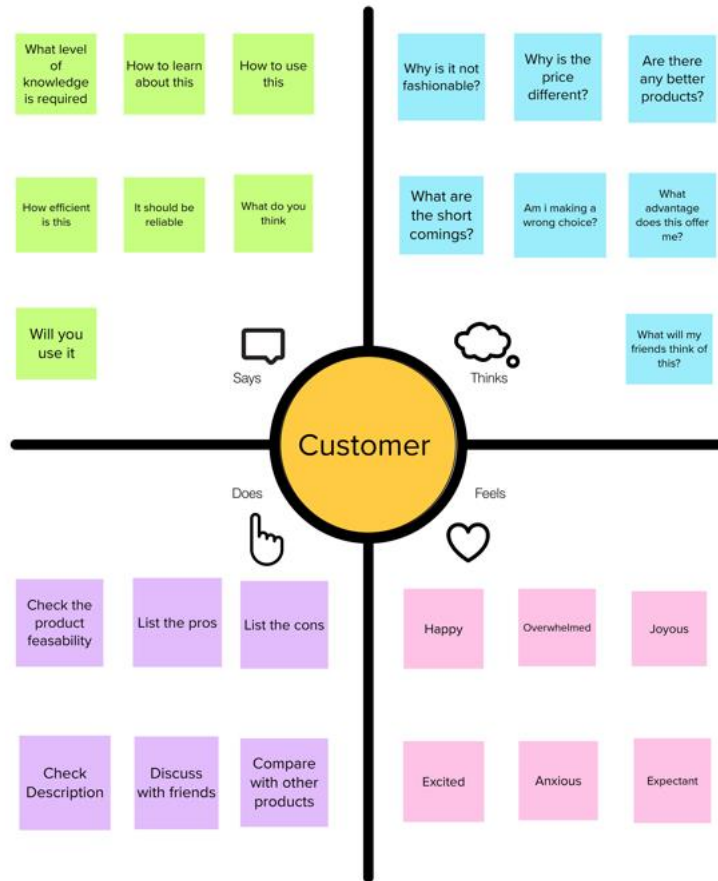
## 2.3 Problem Statement Definition

<b>Problem Statement (PS)</b>	<b>I am (Customer)</b>	<b>I'm trying to</b>	<b>But</b>	<b>Because</b>	<b>Which makes me feel</b>
PS-1	A doctor	Browse the radiology images inside the operation room	It's the highest chance to spread infection	Using the same hand for operation as well as to touch the keyboard, mouse, screen etc	Not sterile
PS-2	A doctor	Browse the radiology images inside the operation room	It affects the patient	Of the noise produced during the voice recognition	Annoying

## CHAPTER 3


### IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas



## 3.2 Ideation & Brainstorming

### Step-1: Team Gathering, Collaboration and Select the Problem Statement



## Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

🕒 10 minutes to prepare  
🕒 1 hour to collaborate  
👤 2-8 people recommended

[Share template feedback](#)

**Before you collaborate**

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

🕒 10 minutes

---

**A Team gathering**  
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

**B Set the goal**  
Think about the problem you'll be focusing on solving in the brainstorming session.

**C Learn how to use the facilitation tools**  
Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →

**1 Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

---

**PROBLEM**  
How might we [your problem statement]?

It's a doctor-computer interaction device in operation room to support medical images manipulation by allowing doctor's hand to maintain sterile, focus by providing faster response.

**Key rules of brainstorming**  
To run a smooth and productive session

- Stay in topic.
- Defer judgment.
- Go for volume.
- Encourage wild ideas.
- Listen to others.
- If possible, be visual.

**Need some inspiration?**  
See a finished version of this template to kickstart your work.

[Open example](#) →



## Step-2: Brainstorm, Idea Listing and Grouping

2

### Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can collect a sticky note and fill the pencil (pen) to make it look like your drawing!

Isha Ambika

Building an algorithm that adapts to all sorts of variation

Perform various operations based on the gestures

Ability to differentiate between different gestures

Ria Kishore

Calculating the distance that contains various hand gestures

Usage of gesture based tool to avoid spreading of infections

Performing various functions for different hand gestures

Bhavet Srivastava

Providing easy interaction between the surgeon and EMR

Reducing the problems related with noise in operation rooms

Remotely accessing the main control with using hand gestures

Jayadevan

Reducing the time involved in controlling the EMR

It replaces the current interface technology with more natural interfaces

Change in focus of attention of surgeon is prevented

3

### Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

10 minutes

TIP

Add color-coded tags to sticky notes to make a cluster that makes sense, organize, and categorize important ideas as they arise within your team.

Ability to differentiate between different gestures

Providing easy interaction between the surgeon and EMR

Reducing the problems related with noise in operation rooms

Building an algorithm that adapts to all sorts of variation

Performing various functions for different hand gestures

Reducing the time involved in controlling the EMR

Change in focus of attention of surgeon is prevented

Collecting the dataset that contains various hand gestures

Perform various operations based on the gestures

Remotely accessing the main control wall using hand gestures

It replaces the current interface technology with more natural interfaces

Usage of gesture based tool to avoid spreading of infections.

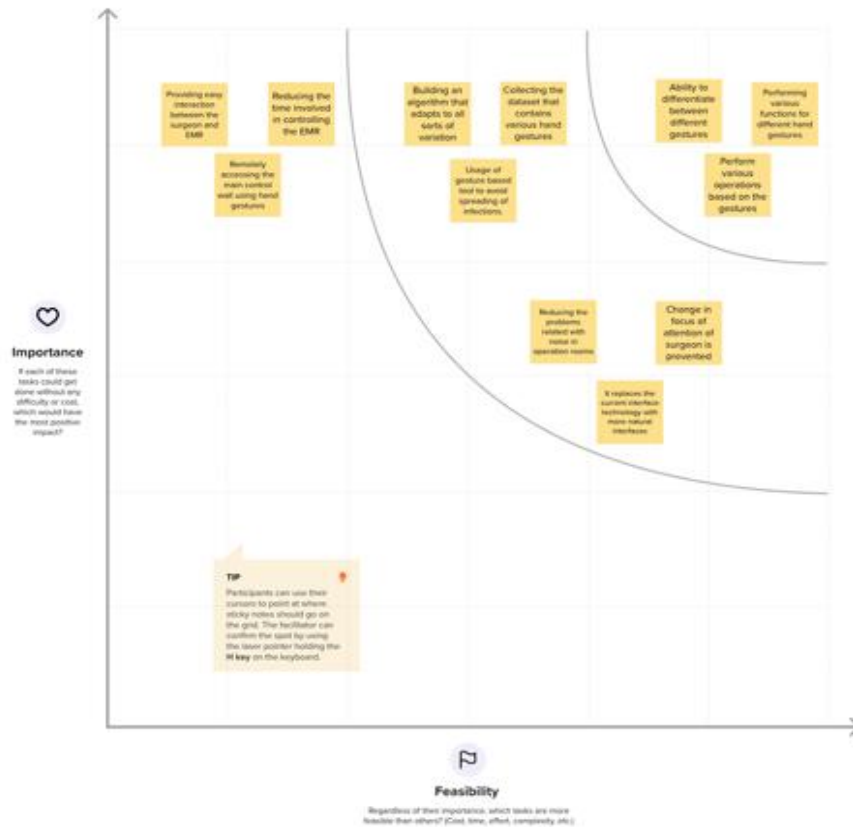
## Step-3: Idea Prioritization

4

### Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



→

### After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

### Quick add-ons

+

#### Share the mural

Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

+

#### Export the mural

Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

### Keep moving forward



#### Strategy blueprint

Define the components of a new idea or strategy.

[Open the template →](#)



#### Customer experience journey map

Understand customer needs, motivations, and obstacles for an experience.

[Open the template →](#)



#### Strengths, weaknesses, opportunities & threats

Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

[Open the template →](#)

[Share template feedback](#)

### 3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A Gesture-Based Tool For Sterile Browsing Of Radiology Images.
2.	Idea / Solution description	The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the model predicts the gestures, then image is blurred , resized & rotated based on the gestures.
3.	Novelty / Uniqueness	Sterile detection method for Radiology Method browsing.
4.	Social Impact / Customer Satisfaction	Formulates more hygiene practices and makes it more easy to browse and upload radiology images. Simplifies
5.	Business Model (Revenue Model)	This comes under the medical field , thus Government allowance and concessions can be expected and the main target user are doctors specialising in radiology related work and multispecialty hospitals
6.	Scalability of the Solution	The structure used for this solution can also be applied to other devices and applications to make them contactless and more smart.

### 3.4 Problem Solution fit

Project Title: **A Gesture-based Tool for Sterile Browsing of Radiology Images**

Project Design Phase-I - **Solution Fit Template**

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> Who is your customer? <b>Used by doctors at hospitals and other medical clinics to avoid contact with infected tools.</b> <b>Used by workers in car manufacturing companies.</b>	<b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span> What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices. <b>Power consumptions needs to be reduced.</b> <b>Customer needs to remember various gestures to use it in appropriate situations.</b> <b>Proper camera to capture the gestures correctly.</b> <b>Stable connection is required to run the software.</b>	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking <b>Doctors can take the tool in their hand, which may cause infections to them.</b> <b>Doctors can use monitors, keyboard but this may lead to inaccurate observations as the doctor will be in movement and this may also cause infections to doctors.</b>	Explore AS, differentiate
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span> Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <b>System helps the customers to show gestures based on which corresponding tools are taken by browsing with radiology images to avoid customers coming in contact.</b>	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations. <b>Unclear images detected by camera may lead to undesired results.</b> <b>As each gesture is mapped to tool, the customer needs to remember gestures to choose a tool.</b> <b>These technologies are expensive and may lead to delay in operation theatre.</b>	<b>7. BEHAVIOUR</b> <span>BE</span> i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace) <b>Proper training is provided to customers to use appropriate gestures for tools.</b> <b>Well equipped manual is provided to customers to resolve their problems and doubts.</b>	
Identify strong TR & EM	<b>3. TRIGGERS</b> <span>TR</span> What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. <b>Technological development in AI and medical industry helps the customers to avoid physical contact.</b>	<b>10. YOUR SOLUTION</b> <span>SL</span> If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. <b>This solution helps the doctors to use gestures to select a tool and to perform operations at faster rate and in efficient way.</b> <b>This solution also avoids doctors coming in physical contact from the infected tools.</b>	<b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span> <b>8.1 ONLINE</b> What kind of actions do customers take online? Extract online channels from #7 <b>Network connection is required to analyze and to choose the tool from captured radio images</b> <b>8.2 OFFLINE</b> What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. <b>Doctors need to use proper gestures for choosing a tool.</b> <b>Power needs to be ava</b>	Identify strong TR & EM
	<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design. <b>Customers feel more safe and secure by using this technology as it prevents them from infections.</b>			

## CHAPTER 4

### REQUIREMENT ANALYSIS

#### 4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Identifying User Gestures	Camera captures the gestures of the user Identification of gestures using the images
FR-2	Deployment in Cloud	Deep Learning model is trained The model is deployed in cloud
FR-3	User Interface	The user interface, which helps in the Human Computer Interaction is designed
FR-4	Gestures related to the Application Domain	The model should be trained with the gestures related to the application domain

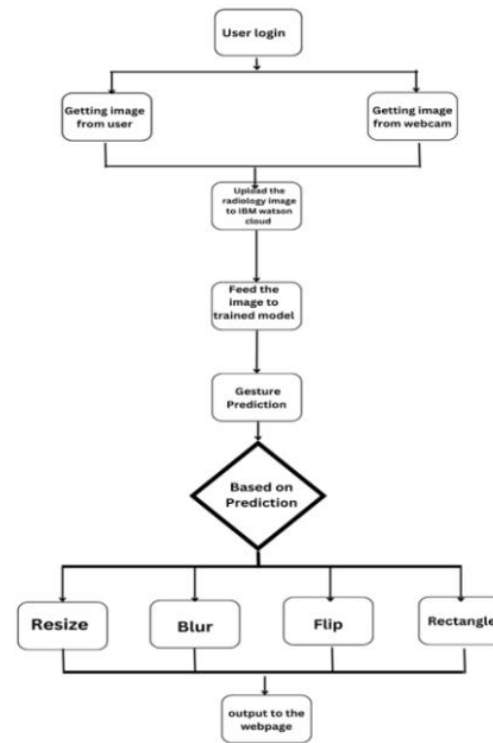
#### 4.2 Non-functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	The user interface which acts as an intermediate between the user and the DL Model which is deployed in the cloud
NFR-2	<b>Security</b>	The model deployed in the cloud should be accessible only by the approved users and it should be inaccessible by the attackers or the terrorists
NFR-3	<b>Reliability</b>	The tool or the system is 95% reliability for a year
NFR-4	<b>Performance</b>	The tool or the system should respond with the accurate response within 4-5 seconds
NFR-5	<b>Availability</b>	The model deployed in the cloud must be available to 99.8% of the people over a month during working hours
NFR-6	<b>Scalability</b>	The model deployed in the cloud must be accessible by over 10,00,000 people trying to access it using the user interface

## CHAPTER 5

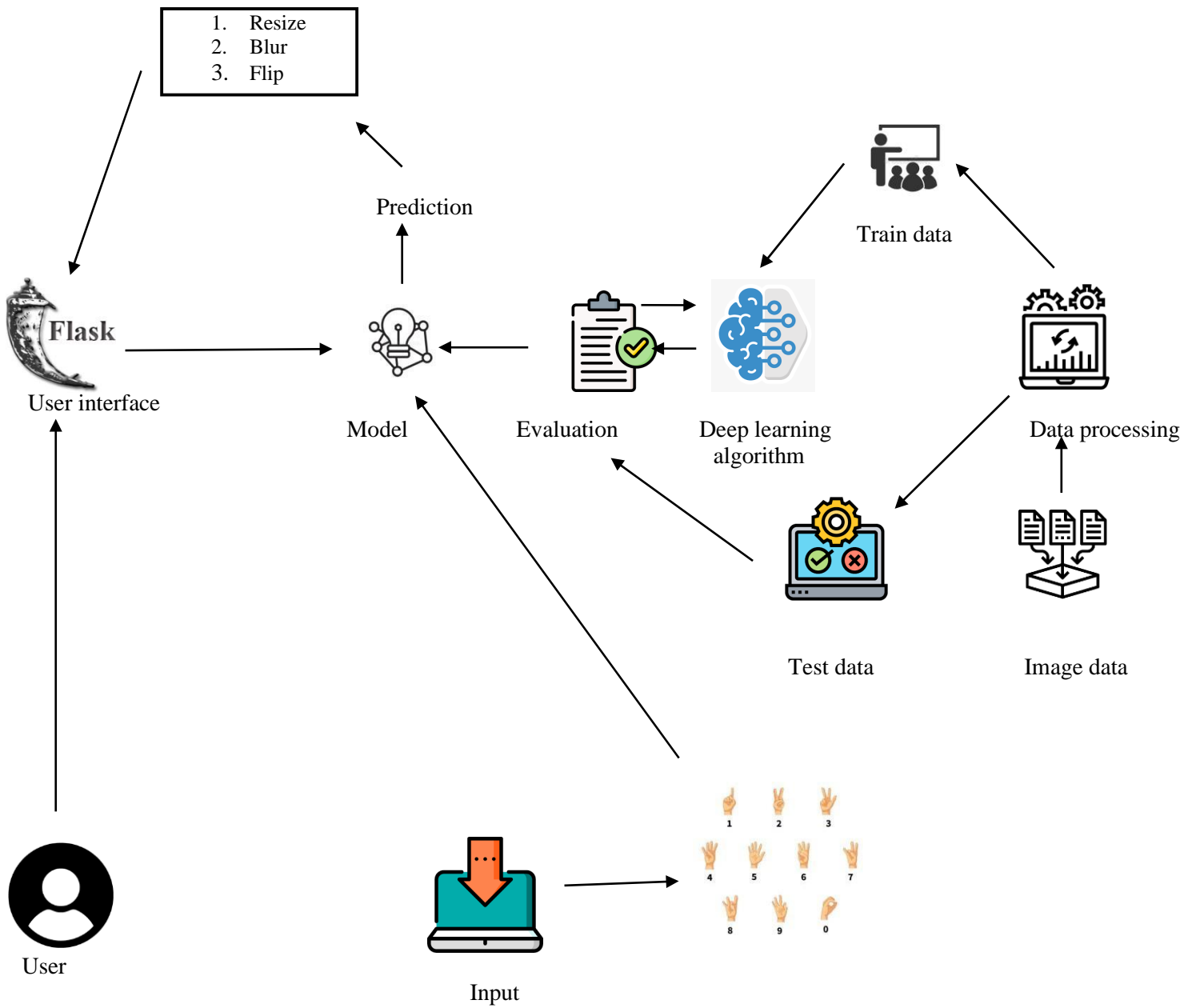
### PROJECT DESIGN

#### 5.1 Data flow diagrams

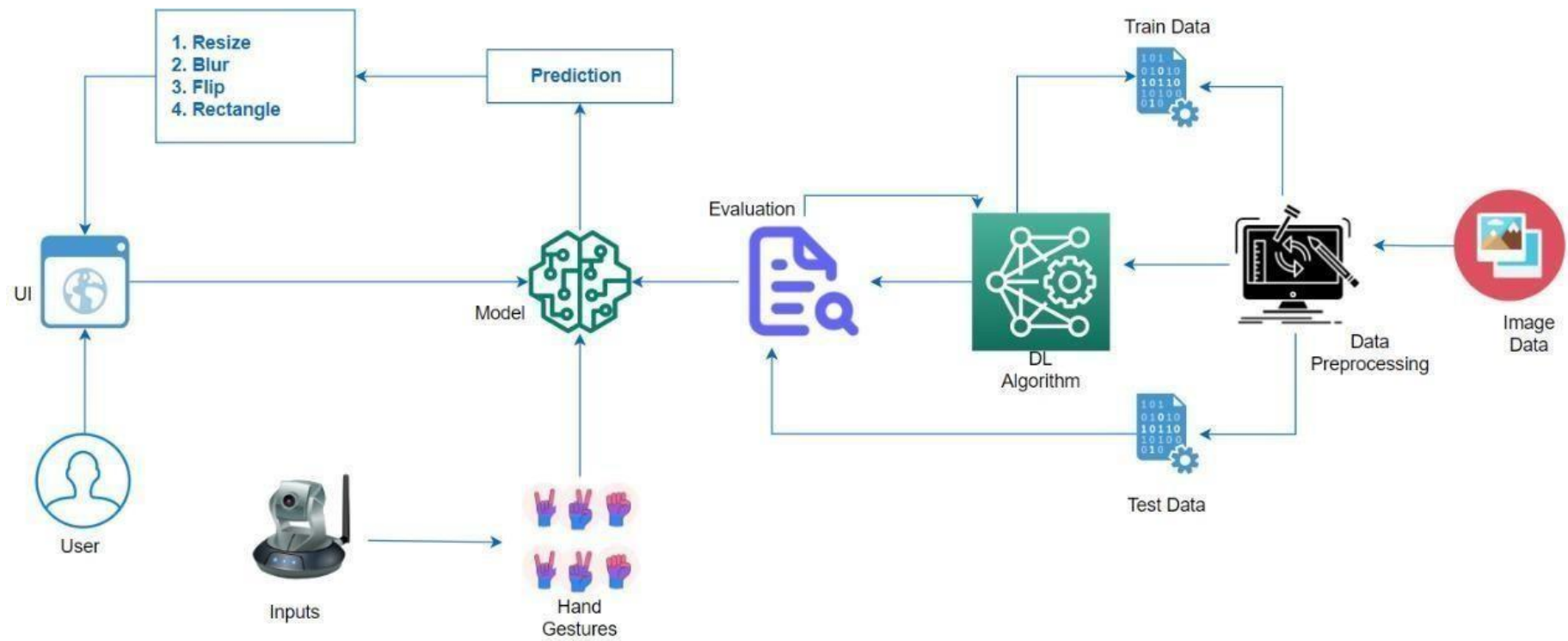


Data Flow Diagram

## 5.2 Solution Architecture



## Technical Architecture





**Table-1 : Components & Technologies:**

<b>S.No</b>	<b>Component</b>	<b>Description</b>	<b>Technology</b>
1.	User Interface	UI ( Web )	HTML, CSS, JavaScript.
2.	Application Logic-1 Image Pre-processing	Input image is pre-processed with the help of library files.	Python, TensorFlow
3.	Application Logic-2Building Model	Building CNN model to recognize the gesture.	Python, Keras
4.	Application Logic-3Creation of app	App is built to obtain gesture as input and to provide as output.	HTML, CSS, JavaScript
5.	Dataset	Hand gesture data set.	From IBM
6.	Cloud Database	User input image is stored in cloud.	IBM Cloud
7.	File Storage	File storage contains dataset and source code.	Device or Drive
8.	Machine Learning Model	CNN Model was used to recognize the pre-processed image by image capturing or by video segmenting.	CNN Model by Python, Keras

**Table-2: Application Characteristics:**

<b>S.No</b>	<b>Characteristics</b>	<b>Description</b>	<b>Technology</b>
1.	Open-Source Frameworks	Application development, data pre-processing.	Visual studio code, Anaconda navigator, TensorFlow
2.	Security Implementations	It identifies the gesture only when the hand is in front of the camera.	OpenCV
3.	Scalable Architecture	It can be used in any environment and is able to identify the gesture	OpenCV
4.	Availability	It is used to reduce the possibility of spreading infections	AI
5.	Performance	Rapid response to the gesture.	CNN

### 5.3 User Stories

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Login	USN-1	Entering Webpage	Enter the application	High	Sprint 1
	Homepage	USN-2	Entering to the “Homepage” of theUI(Webpage)	Enter the homepage	High	Sprint 1
	About	USN-3	I can click on the “About” to details about theApplication	Get the details about theapplication	Low	Sprint 2
	Begin	USN-4	As a user I can upload my radiology imagefrom the computer.	Choose any imagefrom my device	High	Sprint 2
	Predict	USN-5	As a user I can turn on the camera usingpredictbutton	Turn on the camera forprediction	High	Sprint 3
		USN-6	Predicting the images using Hand Gesture	Can resize, blur, and flip my image using my handgesture	High	Sprint 3
		USN-7	I can give a gesture of raised fist and itrecognizes	Can get my fixed resizedimage	High	Sprint 4
		USN-8	I can show my index finger	Can get a rectangularimage	High	Sprint 4
		USN-9	I can show my index finger, middle finger andringfinger at once	Can get my image blurred	High	Sprint 4

## CHAPTER 6

### PROJECT PLANNING & SCHEDULING

#### 6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	TM – 1 TM – 4
Sprint-1	Login	USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	TM – 2 TM – 3
Sprint-2	Dashboard	USN-3	As a user, I can register for the application through Facebook	10	Low	TM – 1 TM – 2
Sprint-1	Details about	USN-4	As a user, I can register for the application through Gmail	5	Medium	TM – 3 TM – 1
Sprint-1	Login and repeated usage	USN-5	As a user, I can log into the application by entering email & password	5	High	TM – 2 TM – 4
Sprint - 2	web page details	USN-6	As a user I must capture images of hand and upload it into the web portal.	10	High	TM – 1 TM – 3
Sprint - 3	Upload the image in the web application	USN-7	As a user I must receive a correct hand gesture as output	20	High	TM – 1 TM – 2
Sprint - 4	Provide efficient customer support	USN-8	As a user, I need to get support from developers in case of queries and failure of service provided	10	Medium	TM – 3 TM – 4
Sprint - 4	Overview the entire process. Take all the responsibility and act bridge between users and developers	USN-9	We need to satisfy the customer needs in an efficient way and make sure any sort of errors are fixed	10	High	TM – 2 TM – 1

## 6.2 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	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

### Velocity:

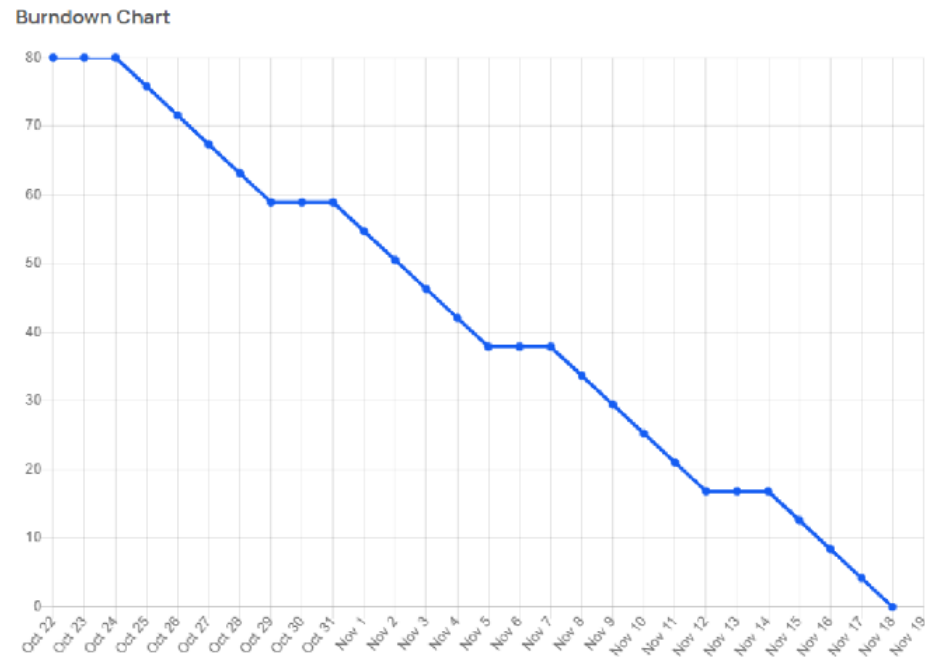
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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

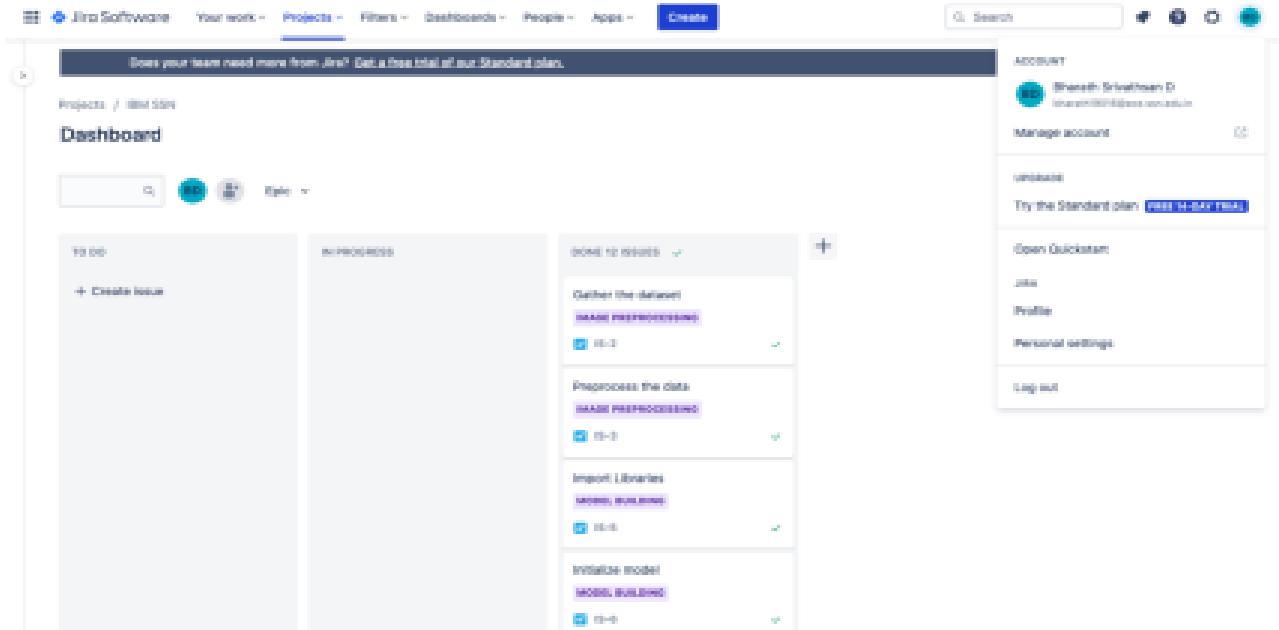
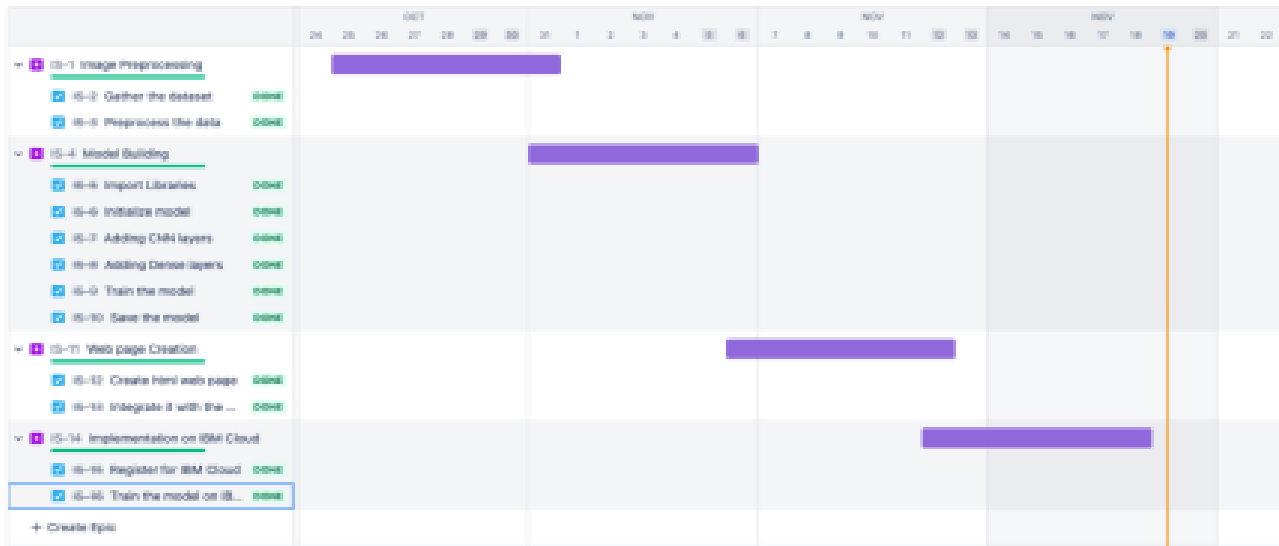
$$AV = \text{sprint duration} / \text{velocity} = 20/6 = 3.33$$

### Burn down Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



## 6.3 Reports from JIRA



## CHAPTER 7

### CODING AND SOLUTIONING

```
from flask import *
from datetime import date
import ibm_db
from flask import Flask, render_template, request
import numpy as np
import operator
import cv2 # opencv library
import os

from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
from werkzeug.utils import secure_filename

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

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;PROTOCOL=TCPIP;UID=mml48790;PWD=dMMqquglkWltg4Jo", "", "")

print(conn)

print("Connecting Successful.....")

user_id=0

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



```

        return render_template("index.html",status="",colour="red")
elif request.method=='POST':
    global retailer_id
    email=request.form["email"]
    password=request.form["password"]
    query = "select * from user where email = '{ }'".format(email)
    exec_query = ibm_db.exec_immediate(conn, query)
    row = ibm_db.fetch_both(exec_query)
    if(row is not False):
        if(row['PASSWORD'] != password):
            return render_template("index.html",status="Invalid Password",colour="red")
        else:
            temp="select id from user where email = '{ }'".format(email)
            exec_query = ibm_db.exec_immediate(conn, temp)
            dict= ibm_db.fetch_both(exec_query)
            user_id=dict["ID"]
            return render_template("home.html")

    return render_template("index.html",status="Invalid Email",colour="red")

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

    if request.method=='GET':
        return render_template("register.html",status="",colour="red")
    elif request.method=='POST':
        email=request.form["email"]

```

```

password=request.form["password"]
username=request.form["username"]
query = "select * from user where email = '{ }'".format(email)
exec_query = ibm_db.exec_immediate(conn, query)
row = ibm_db.fetch_both(exec_query)

if(row is False):
    query = "insert into user(email, password, username) values('{ }', '{ }', '{ }')".format(email,
password, username)
    exec_query = ibm_db.exec_immediate(conn, query)
    return render_template("index.html",status="Signup Success",colour="green")
else:
    return render_template("register.html",status="User Already Exists",colour="red")

@app.route('/home') # routes to the intro page
def home():
    return render_template('home.html') #rendering the intro 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 index6():
    return render_template("index6.html")

```

```

@app.route('/predict', methods=['GET', 'POST']) # route to show the predictions in a web UI
def launch():
    #Getting input and storing it
    if request.method == 'POST':
        print('inside launch function')
        f=request.files['image']

        basepath=os.path.dirname(_file_)
        file_path=os.path.join(basepath,'uploads',secure_filename(f.filename))
        f.save(file_path)
        print('img saved successfully')
        print(file_path)
        # test_image=cv2.imread(file_path,cv2.IMREAD_COLOR)
        # test_image=cv2.resize(test_image,(64,64))
        # result= model.predict(test_image.reshape(1,64,64,1))

        # img = load_img(file_path, grayscale=True, target_size=(64, 64))
        # x = img_to_array(img)
        # x = np.expand_dims(x, axis = 0)

    cap=cv2.VideoCapture(0)
    image1=cv2.imread(file_path)
    cv2.imshow("Output",image1)
    prev='NULL'
    while True:
        _, frame=cap.read()
        frame=cv2.flip(frame,1)

```

```

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

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

roi=cv2.resize(roi,(64,64))
roi=cv2.cvtColor(roi,cv2.COLOR_BGR2GRAY)
_, test_image=cv2.threshold(roi,120,255,cv2.THRESH_BINARY)
##cv2.imshow("test",test_image)

result = model.predict(test_image.reshape(1,64,64,1))
print(result)

prediction =
{'ZERO':result[0][0],'ONE':result[0][1],'TWO':result[0][2],'THREE':result[0][3],'FOUR':result[0][4],'FIVE':result[0][5]}

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

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

interrupt=cv2.waitKey(10)

if interrupt & 0xFF == 27: #Esc key to break from the while loop
    break

```

```

if prev == prediction:
    continue

prev = prediction

image1=cv2.imread(file_path)
image1=cv2.resize(image1,(255,255))
if prediction[0][0]=='ONE':
    resized=cv2.resize(image1,(200,200))
    cv2.destroyWindow("Output")
    cv2.imshow("Output",resized)
##    key=cv2.waitKey(3000)
##
##    if(key & 0xFF) == ord("1"):
##        cv2.destroyWindow("Fixed Resizing - One")

elif prediction[0][0]=='ZERO':
    cv2.rectangle(image1,(480,170),(650,420),(0,0,255),2)
    cv2.destroyWindow("Output")
    cv2.imshow("Output",image1)
    #cv2.imshow("Rectangle - Zero",image1)
    #cv2.waitKey(0)
##    key=cv2.waitKey(3000)
##
##    if(key & 0xFF)==ord("0"):

```

```

##         cv2.destroyAllWindows("Rectangle - Zero")

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.destroyAllWindows("Output")

    cv2.imshow("Output",rotated)

##         cv2.imshow("OpenCV Rotation - Two",rotated)

##         key=cv2.waitKey(3000)

##         if(key & 0xFF)==ord("2"):

##             cv2.destroyAllWindows("OpenCV Rotation - Two")

```

```

elif prediction[0][0]=='THREE':

    blurred=cv2.GaussianBlur(image1,(11,11),0)

    cv2.destroyAllWindows("Output")

    cv2.imshow("Output",blurred)

##         cv2.imshow("Blurred - Three",blurred)

##         key=cv2.waitKey(3000)

##         if(key & 0xFF)==ord("3"):

##             cv2.destroyAllWindows("Blurred - Three")

```

```

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

    zoomed=cv2.resize(image1,(400,400))

    cv2.destroyAllWindows("Output")

    cv2.imshow("Output",zoomed)

```

```

##      cv2.imshow("Zoomed - Four",zoomed)
##      key=cv2.waitKey(3000)
##      if(key & 0xFF)==ord("4"):
##          cv2.destroyWindow("Zoomed - Four")

elif prediction[0][0]=='FIVE':
    neg=255-image1
    cv2.destroyWindow("Output")
    cv2.imshow("Output",neg)
##      cv2.imshow("Negative - Five",neg)
##      key=cv2.waitKey(3000)
##      if(key & 0xFF)==ord("5"):
##          cv2.destroyWindow("Negative - Five")

else:
    continue

cap.release()
cv2.destroyAllWindows()

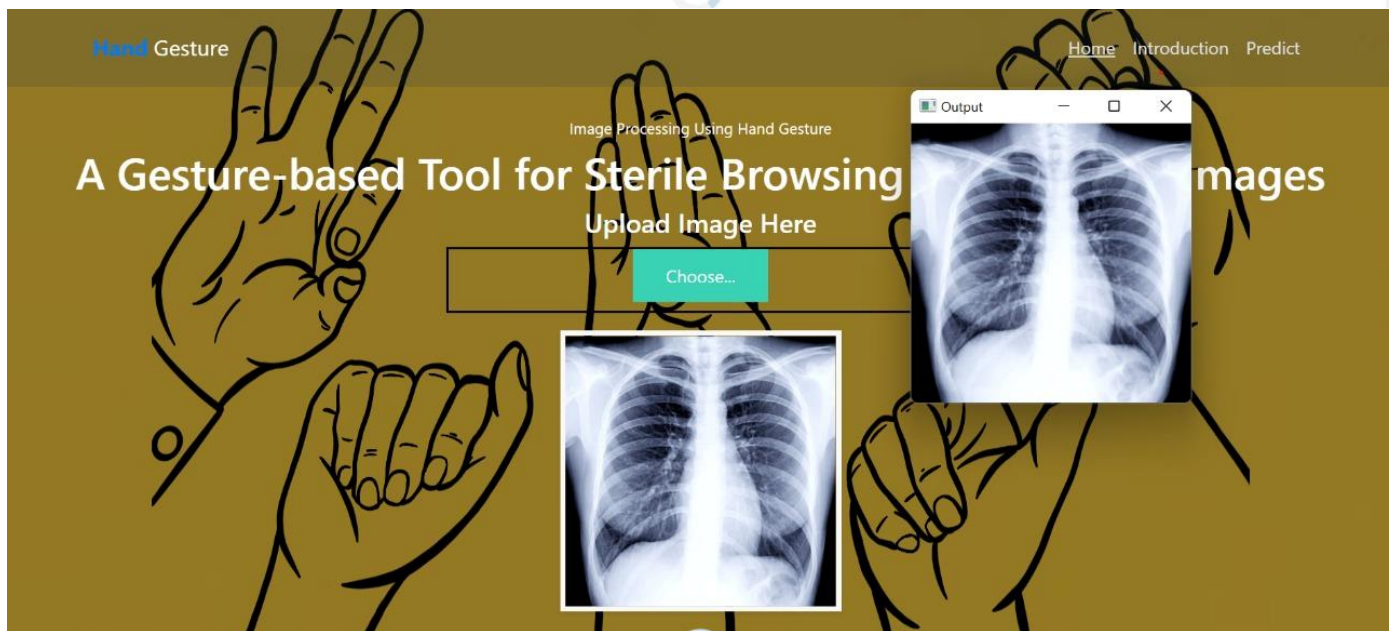
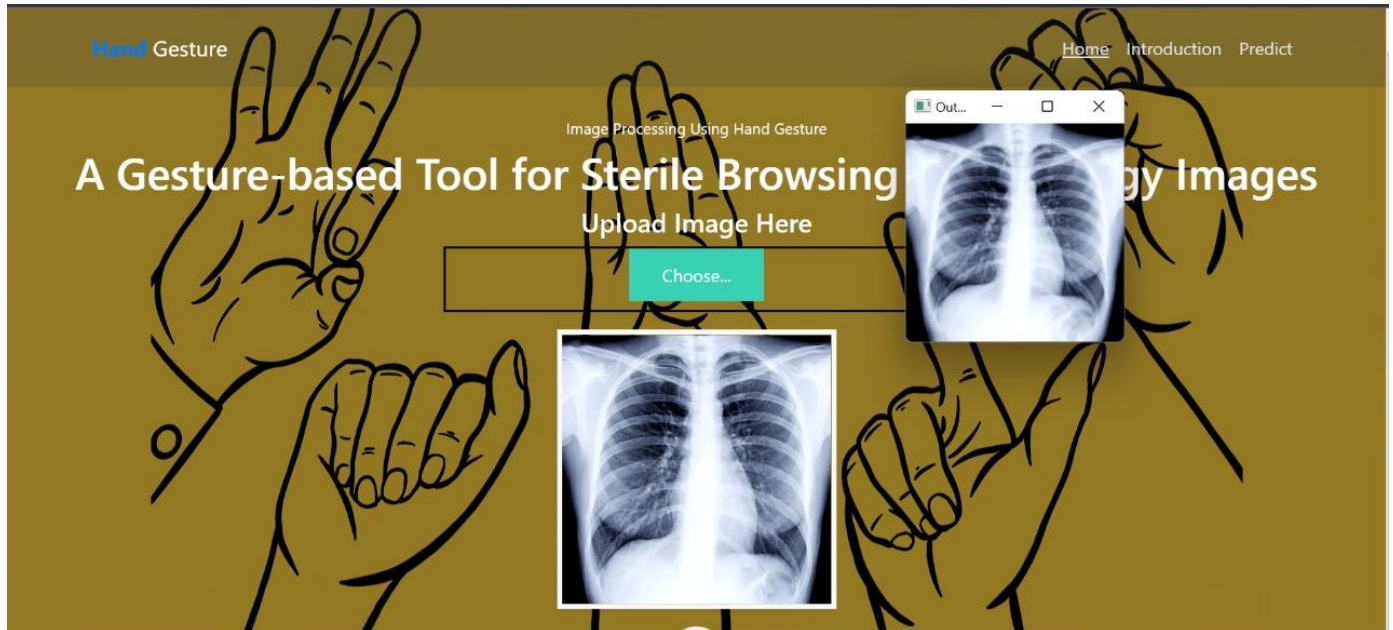
return render_template("home.html")

if __name__=="main_":
    app.run(host='0.0.0.0',debug=True)

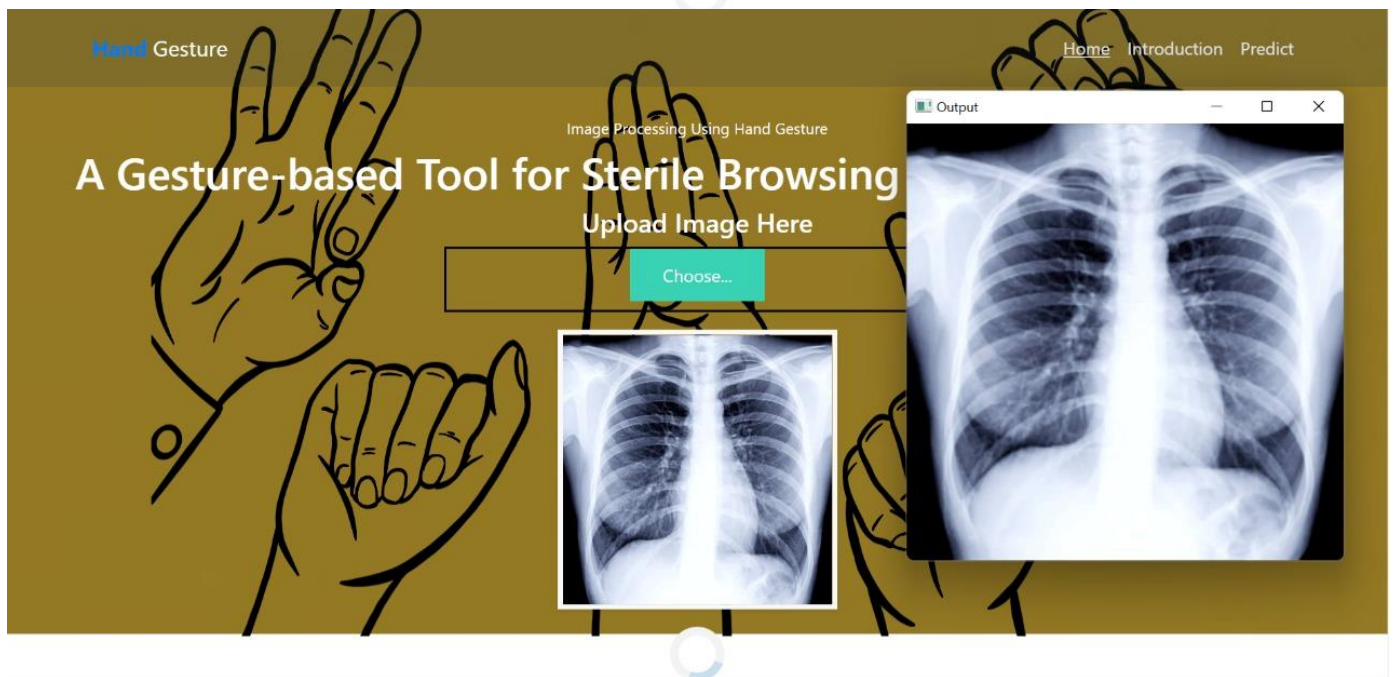
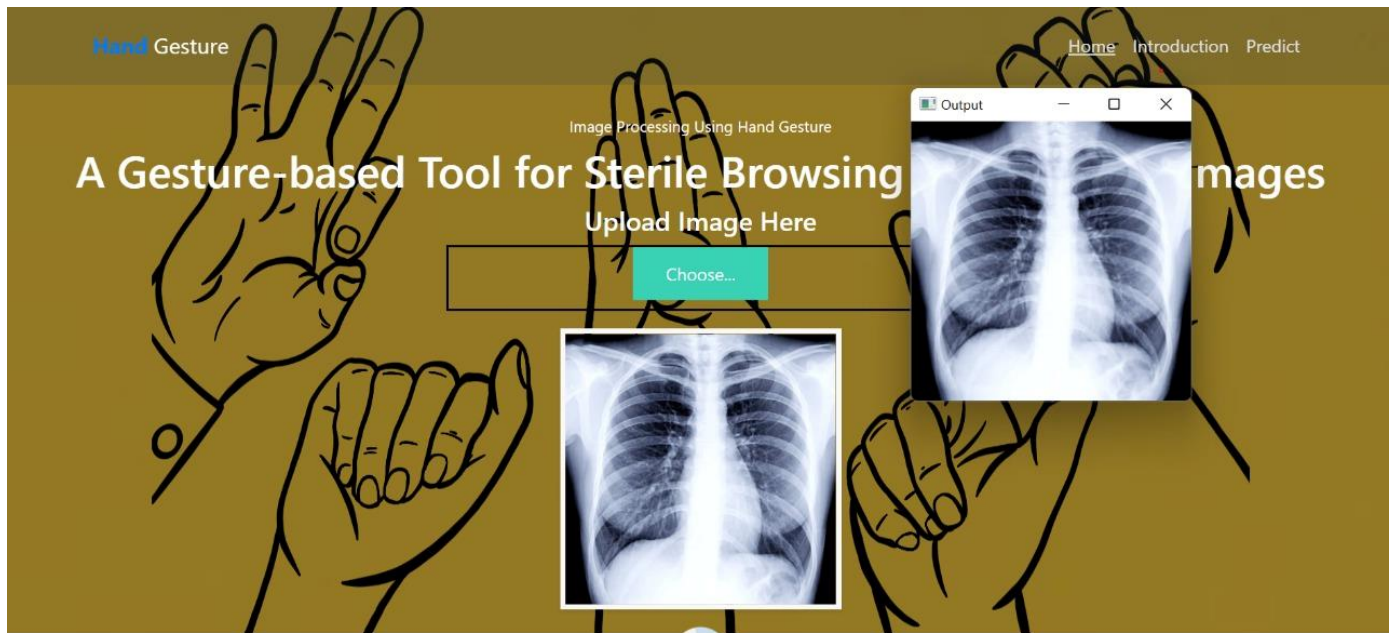
```

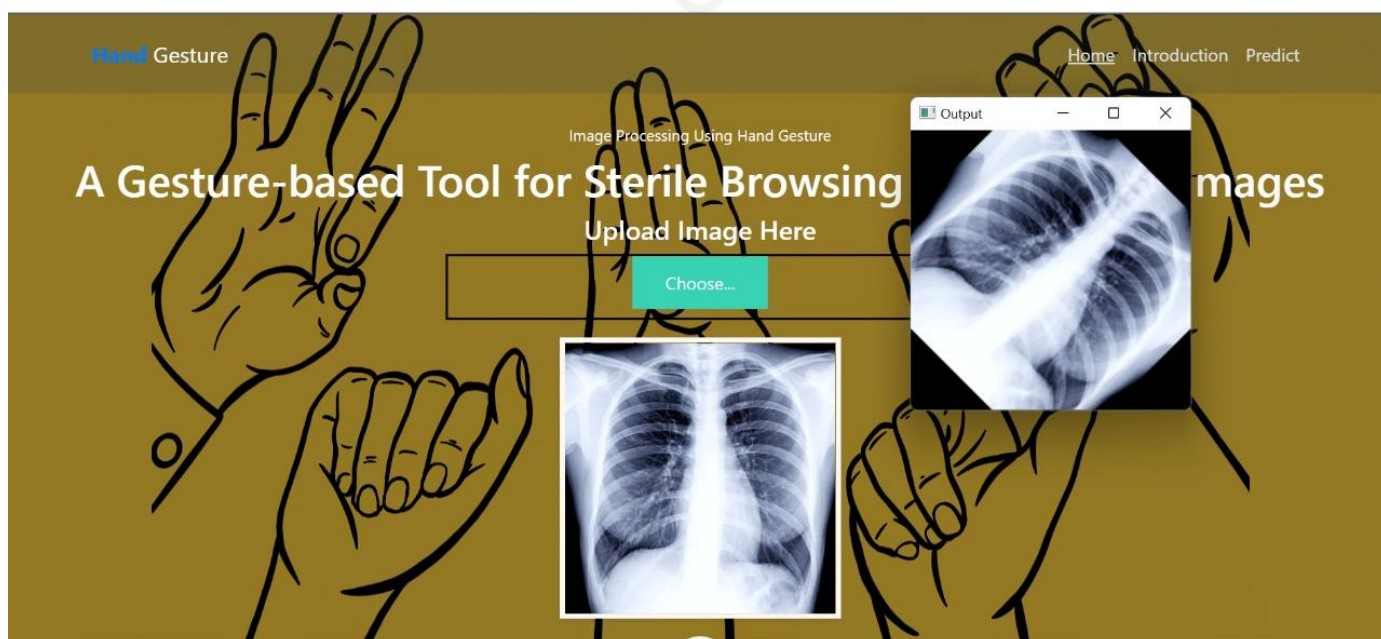
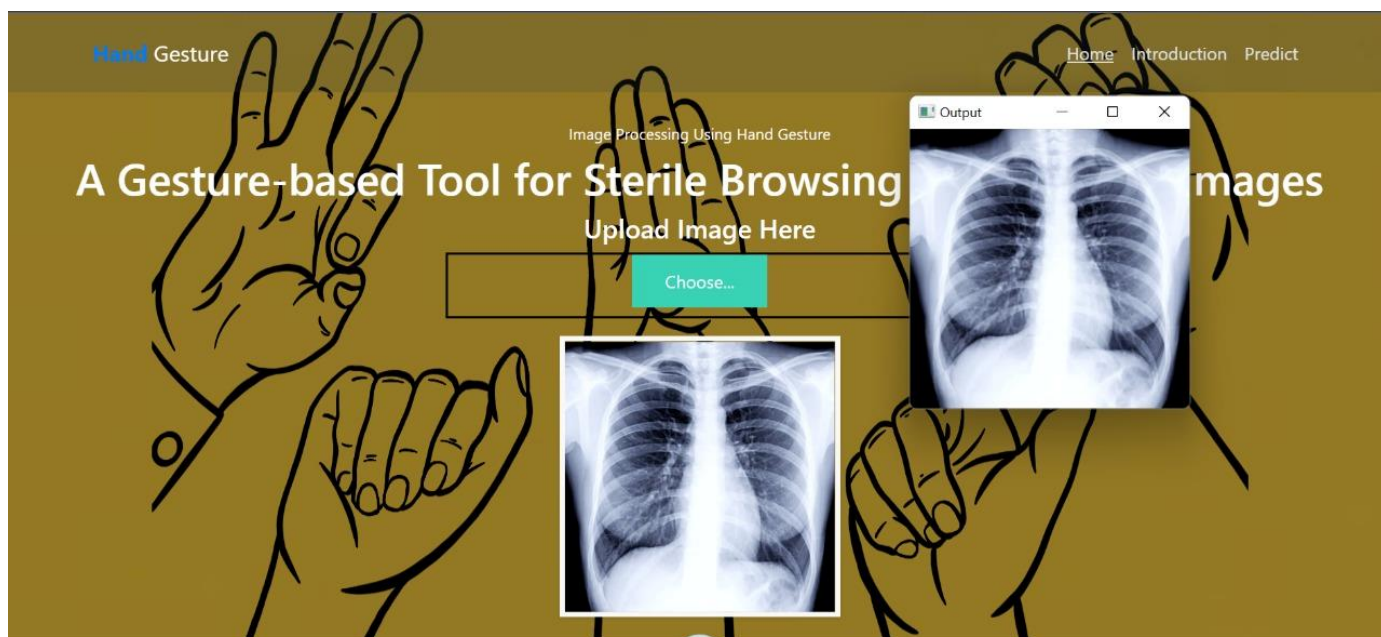
## CHAPTER 8

### TESTING







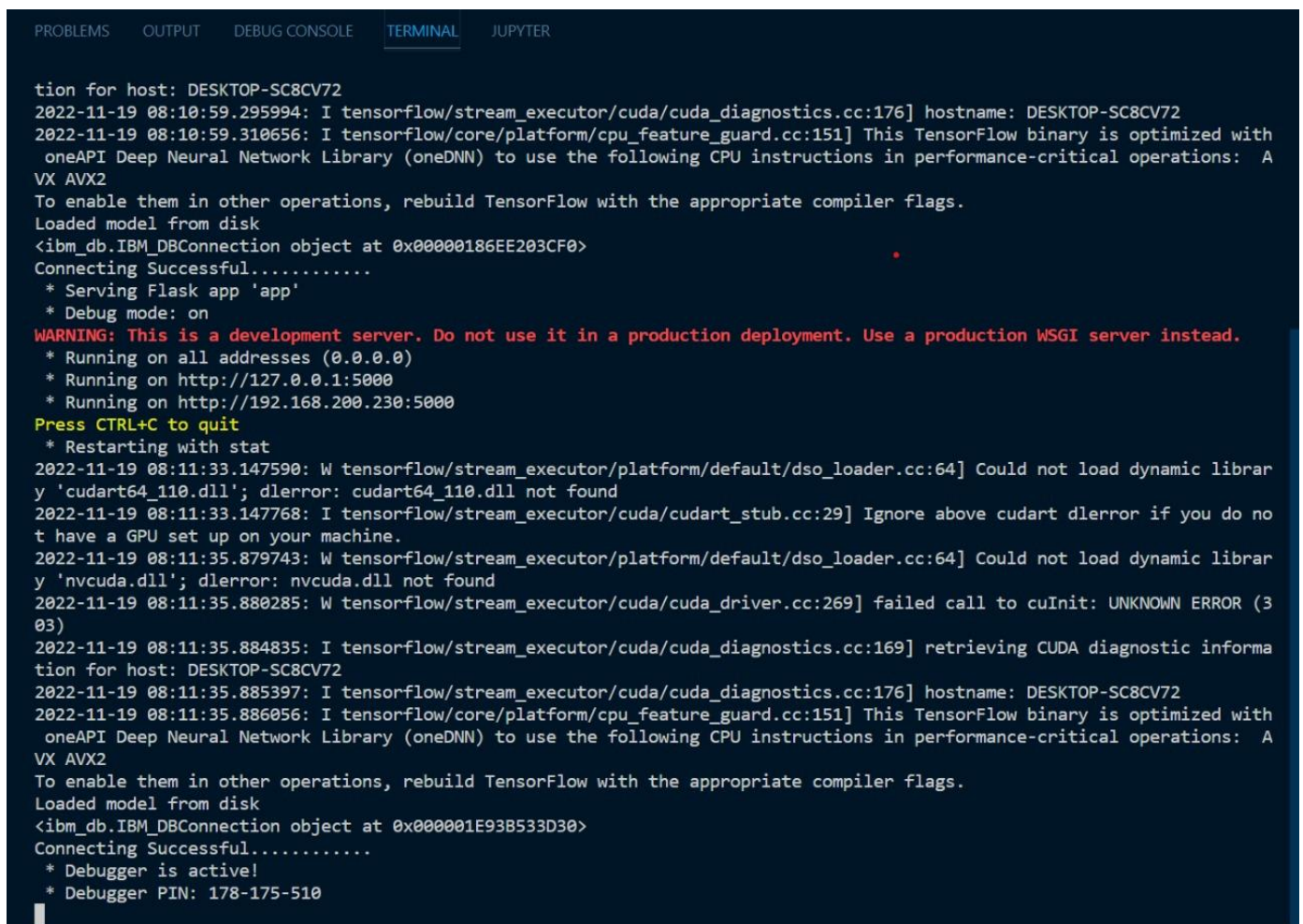


## CHAPTER 9

### RESULTS

In this project, we found that we can maintain the sterility of an operation theater, etc by using hand based gesture tools to browse the images obtained.

Final findings (Output) of the project along with screenshots as follows.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

tion for host: DESKTOP-SC8CV72
2022-11-19 08:10:59.295994: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP-SC8CV72
2022-11-19 08:10:59.310656: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with
oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: A
VX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Loaded model from disk
<ibm_db.IBM_DBConnection object at 0x00000186EE203CF0>
Connecting Successful.....
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.200.230:5000
Press CTRL+C to quit
* Restarting with stat
2022-11-19 08:11:33.147590: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic librar
y 'cudart64_110.dll'; dLError: cudart64_110.dll not found
2022-11-19 08:11:33.147768: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dLError if you do no
t have a GPU set up on your machine.
2022-11-19 08:11:35.879743: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic librar
y 'nvcuda.dll'; dLError: nvcuda.dll not found
2022-11-19 08:11:35.880285: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (3
03)
2022-11-19 08:11:35.884835: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic informa
tion for host: DESKTOP-SC8CV72
2022-11-19 08:11:35.885397: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP-SC8CV72
2022-11-19 08:11:35.886056: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with
oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: A
VX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Loaded model from disk
<ibm_db.IBM_DBConnection object at 0x000001E93B533D30>
Connecting Successful.....
* Debugger is active!
* Debugger PIN: 178-175-510
```



## **CHAPTER 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 process it.

## **CHAPTER 11**

### **CONCLUSION**

In this project we developed a tool which recognizes the hand gestures and enables doctors to browse through radiology images using these gestures. This enables doctors and surgeons to maintain the sterility as they would not have to touch any mouse or keyboard to go through the images. This tool is also easy to use and is quicker than the regular method of using mouse/keyboard. It can be used regardless of the users location since they don't have to be in contact with any device. It also does not require the user to have any device on them to use it. Further this technology can be extended to other industries like it can be used by presenters, by teachers for show images in the classroom, etc.

## **CHAPTER 12**

### **FUTURE SCOPE**

- The tool can be made quicker by increasing the recognition speed.
- More number of gestures can be added thereby increasing this tool's functionality and use ability 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.

## CHAPTER 13

### APPENDIX

#### Source Code:

```
from
flas
k
impo
rt *

from datetime import date
import ibm_db
from flask import Flask, render_template, request
import numpy as np
import operator
import cv2 # opencv library
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
from werkzeug.utils import secure_filename

app = Flask(__name__, template_folder="templates")
model=load_model('gesture.h5')
print("Loaded model from disk")
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-
7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;PROTOCOL=T
CPIP;UID=mm148790;PWD=dMMqquglkWltg4Jo", "", "")
print(conn)
print("Connecting Successful.....")
user_id=0

@app.route("/", methods=['GET','POST'])
def login():
    if request.method=='GET':
        return render_template("index.html",status="",colour="red")
    elif request.method=='POST':
        global retailer_id
        email=request.form["email"]
        password=request.form["password"]
        query = '''select * from user where email = \'{ }\'''.format(email)
        exec_query = ibm_db.exec_immediate(conn, query)
```

```

row = ibm_db.fetch_both(exec_query)
if(row is not False):
    if(row['PASSWORD'] != password):
        return render_template("index.html",status="Invalid Password",colour="red")
    else:
        temp='''select id from user where email = \'{}\'''.format(email)
        exec_query = ibm_db.exec_immediate(conn, temp)
        dict= ibm_db.fetch_both(exec_query)
        user_id=dict["ID"]
        return render_template("home.html")

return render_template("index.html",status="Invalid Email",colour="red")

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

    if request.method=='GET':
        return render_template("register.html",status="",colour="red")
    elif request.method=='POST':
        email=request.form["email"]
        password=request.form["password"]
        username=request.form["username"]
        query = '''select * from user where email = \'{}\'''.format(email)
        exec_query = ibm_db.exec_immediate(conn, query)
        row = ibm_db.fetch_both(exec_query)
        if(row is False):
            query = '''insert into user(email, password, username) values('{}', '{}',
'{}')'''.format(email, password, username)
            exec_query = ibm_db.exec_immediate(conn, query)
            return render_template("index.html",status="Signup Success",colour="green")
        else:
            return render_template("register.html",status="User Already Exists",colour="red")

@app.route('/home') # routes to the intro page
def home():
    return render_template('home.html') #rendering the intro 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 index6():
    return render_template("index6.html")

@app.route('/predict', methods=['GET', 'POST']) # route to show the predictions in a web UI
def launch():
    #Getting input and storing it
    if request.method == 'POST':
        print('inside launch function')
        f=request.files['image']

        basepath=os.path.dirname(__file__)
        file_path=os.path.join(basepath,'uploads',secure_filename(f.filename))
        f.save(file_path)
        print('img saved successfully')
        print(file_path)
        # test_image=cv2.imread(file_path,cv2.IMREAD_COLOR)
        # test_image=cv2.resize(test_image,(64,64))
        # result= model.predict(test_image.reshape(1,64,64,1))

        # img = load_img(file_path, grayscale=True, target_size=(64, 64))
        # x = img_to_array(img)
        # x = np.expand_dims(x, axis = 0)

    cap=cv2.VideoCapture(0)
    image1=cv2.imread(file_path)
    cv2.imshow("Output",image1)
    prev='NULL'
    while True:
        __, frame=cap.read()
        frame=cv2.flip(frame,1)

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

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

        roi=cv2.resize(roi,(64,64))
        roi=cv2.cvtColor(roi,cv2.COLOR_BGR2GRAY)
        __, test_image=cv2.threshold(roi,120,255,cv2.THRESH_BINARY)

```

```

    ##cv2.imshow("test",test_image)

    result = model.predict(test_image.reshape(1,64,64,1))
    print(result)
    prediction =
{'ZERO':result[0][0],'ONE':result[0][1],'TWO':result[0][2],'THREE':result[0][3],'FOUR':result[
0][4],'FIVE':result[0][5]}
    prediction=sorted(prediction.items(),key=operator.itemgetter(1),reverse=True)

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


    interrupt=cv2.waitKey(10)
    if interrupt & 0xFF == 27: #Esc key to break from the while loop
        break

    if prev == prediction:
        continue

    prev = prediction


    image1=cv2.imread(file_path)
    image1=cv2.resize(image1,(255,255))
    if prediction[0][0]=='ONE':
        resized=cv2.resize(image1,(200,200))
        cv2.destroyWindow("Output")
        cv2.imshow("Output",resized)
    ##        key=cv2.waitKey(3000)
    ##
    ##        if(key & 0xFF) == ord("1"):
    ##            cv2.destroyWindow("Fixed Resizing - One")

    elif prediction[0][0]=='ZERO':
        cv2.rectangle(image1,(480,170),(650,420),(0,0,255),2)
        cv2.destroyWindow("Output")
        cv2.imshow("Output",image1)
        #cv2.imshow("Rectangle - Zero",image1)
        #cv2.waitKey(0)
    ##        key=cv2.waitKey(3000)
    ##
    ##        if(key & 0xFF)==ord("0"):

```

```

##             cv2.destroyAllWindows("Rectangle - Zero")

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.destroyAllWindows("Output")
    cv2.imshow("Output",rotated)
##             cv2.imshow("OpenCV Rotation - Two",rotated)
##             key=cv2.waitKey(3000)
##             if(key & 0xFF)==ord("2"):
##                 cv2.destroyAllWindows("OpenCV Rotation - Two")

elif prediction[0][0]=='THREE':
    blurred=cv2.GaussianBlur(image1,(11,11),0)
    cv2.destroyAllWindows("Output")
    cv2.imshow("Output",blurred)
##             cv2.imshow("Blurred - Three",blurred)
##             key=cv2.waitKey(3000)
##             if(key & 0xFF)==ord("3"):
##                 cv2.destroyAllWindows("Blurred - Three")

elif prediction[0][0]=='FOUR':
    zoomed=cv2.resize(image1,(400,400))
    cv2.destroyAllWindows("Output")
    cv2.imshow("Output",zoomed)
##             cv2.imshow("Zoomed - Four",zoomed)
##             key=cv2.waitKey(3000)
##             if(key & 0xFF)==ord("4"):
##                 cv2.destroyAllWindows("Zoomed - Four")

elif prediction[0][0]=='FIVE':
    neg=255-image1
    cv2.destroyAllWindows("Output")
    cv2.imshow("Output",neg)
##             cv2.imshow("Negative - Five",neg)
##             key=cv2.waitKey(3000)
##             if(key & 0xFF)==ord("5"):
##                 cv2.destroyAllWindows("Negative - Five")

else:
    continue

```

```
cap.release()
cv2.destroyAllWindows()

return render_template("home.html")

if __name__=="__main__":
    app.run(host='0.0.0.0',debug=True)
```

## **Github Link:**

<https://github.com/IBM-EPBL/IBM-Project-19498-1659698742>

## **Project Demo Link:**

<https://drive.google.com/file/d/1rwdC6QbhhNBRucegrVSkFI3El48Gbaly/view?usp=sharing>