A GESTURE- BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

(**TEAM ID : PNT2022TMID29207**)

PROJECT REPORT

Submitted by

GOKULNATH K

SANTHOSH KUMAR G

KARAN P

KISHORE KUMAR S

In partial fullfillment for the award of degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 025

INDEX

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. The recently developed depth sensors, e.g., the Kinect sensor, have provided new opportunities for human-computer interaction (HCI). Although great progress has been made by leveraging the Kinect sensor, e.g., in human body tracking, face recognition and human action recognition, robust hand gesture recognition remains an open problem. Compared to the entire human body, the hand is a smaller object with more complex articulations and more easily affected by segmentation errors. It is thus a very challenging problem to recognize hand gestures. This paper focuses on building a robust part-based hand gesture recognition system using Kinect sensor. To handle the noisy hand shapes obtained from the Kinect sensor, we propose a novel distance metric, Finger-Earth Mover's Distance (FEMD), to measure the dissimilarity between hand shapes. As it only matches the finger parts while not the whole hand, it can better distinguish the hand gestures of slight differences. The extensive experiments demonstrate that our hand gesture recognition system is accurate (a 93.2% mean accuracy on a challenging 10gesture dataset), efficient (average 0.0750 s per frame), robust to hand articulations, distortions and orientation or scale changes, and can work in uncontrolled environments (cluttered backgrounds and lighting conditions). The superiority of our system is further demonstrated in two real-life HCI applications.

2. In this paper, we address natural human-robot interaction (HRI) in a smart assisted living (SAIL) system for the elderly and the disabled. Two common HRI problems are studied: hand gesture recognition and daily activity recognition. For hand gesture recognition, we implemented a neural network for gesture spotting and a hierarchical hidden Markov model for context-based recognition. For daily activity recognition, a multi sensor fusion scheme is developed to process motion data collected from the foot and the waist of a human subject. Experiments using a prototype wearable sensor system show the effectiveness and accuracy of our algorithms.

3. The use of doctor-computer interaction devices in the operation room (OR) requires new modalities that support medical imaging manipulation while allowing doctors' hands to remain sterile, supporting their focus of attention, and providing fast response times. This paper presents "*Gestix*," a vision-based hand gesture capture and recognition system that interprets in real-time the user's gestures for navigation and manipulation of images in an electronic medical record (EMR) database. Navigation and other gestures are translated

to commands based on their temporal trajectories, through video capture. "*Gestix*" was tested during a brain biopsy procedure. In the in vivo experiment, this interface prevented

the surgeon's focus shift and change of location while achieving a rapid intuitive reaction and easy interaction. Data from two usability tests provide insights and implications regarding human-computer interaction based on nonverbal conversational modalities.

4. They have developed a system that uses computer vision to replace standard computer mouse functions with hand gestures. The system is designed to enable non-contact human-computer interaction (HCI), so that surgeons will be able to make more effective use of computers during surgery. In this paper, They begin by discussing the need for noncontact computer interfaces in the operating room. They then describe the design of our non-contact mouse system, focusing on the techniques used for hand detection, tracking, and gesture recognition. Finally, They present preliminary results from testing and planned future work.

5. 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 manipulate objects in an image visualization environment. A color distribution model of the gamut of colors of the users hand or glove is built at the start of each session resulting in an independent system. The gesture system relies on realtime robust tracking of the user's hand based on a color-motion fusion model, in which the relative weight applied to the motion and color cues are adaptively determined according to the state of the system. Dynamic navigation gestures are translated to commands based on their relative positions on the screen. A state machine switches between other gestures such as zoom and rotate, as well as a sleep state. Performance evaluation included gesture recognition accuracy, task learning, and rotation accuracy. Fast task learning rates were found with convergence after ten trials. A beta test of a system prototype was conducted during a live brain biopsy operation, where neurosurgeons were able to browse through MRI images of the patient's brain using the sterile hand gesture interface. The surgeons indicated the system was easy to use and fast with high overall satisfaction.

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. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor, August 2013, IEEE Transactions on Multimedia 15(5):1110-1120; DOI:10.1109/TMM.2013.2246148
2. Wearable Sensor-Based Hand Gesture and Daily Activity Recognition for Robot- Assisted LivingIEEE Transactionson systems, Man , andCybernetics Part A: Systemsand humans, Vol .41, No. 3, May 2011.
3. A Gesture-based Tool for Sterile Browsing of Radiology Image , J Am Med Inform Assoc. 2008 May-Jun; 15(3): 321–323, doi: 10.1197/jamia.M241.
4. A Non-Contact Mouse for Surgeon-Computer Interaction, February 2004, Technology and health care: official journal of the European Society for Engineering and Medicine 12(3):245-57; DOI:10.3233/THC-2004-12304.
5. Real-Time Hand Gesture Interface for Browsing Medical Images, February 2007; International Journal of Intelligent Computing in Medical Sciences and Image Processing 2(1); DOI:10.1080/1931308X.2008.10644149.

2.3 Problem Statement Definition

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
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 noice produced during the voice recognition	Annoying

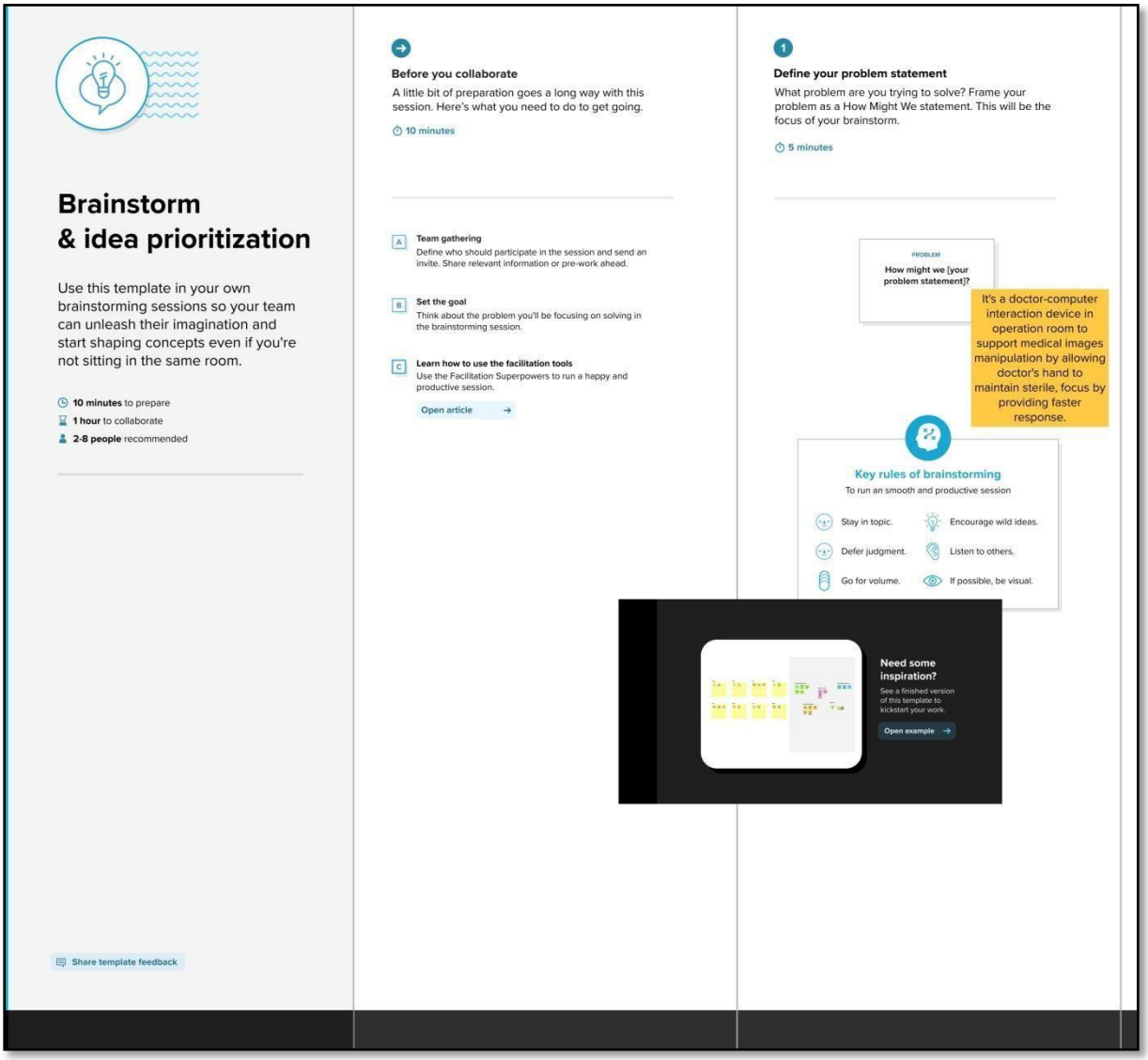
CHAPTER 3

IDEATION & PROPOSED SOLUTION

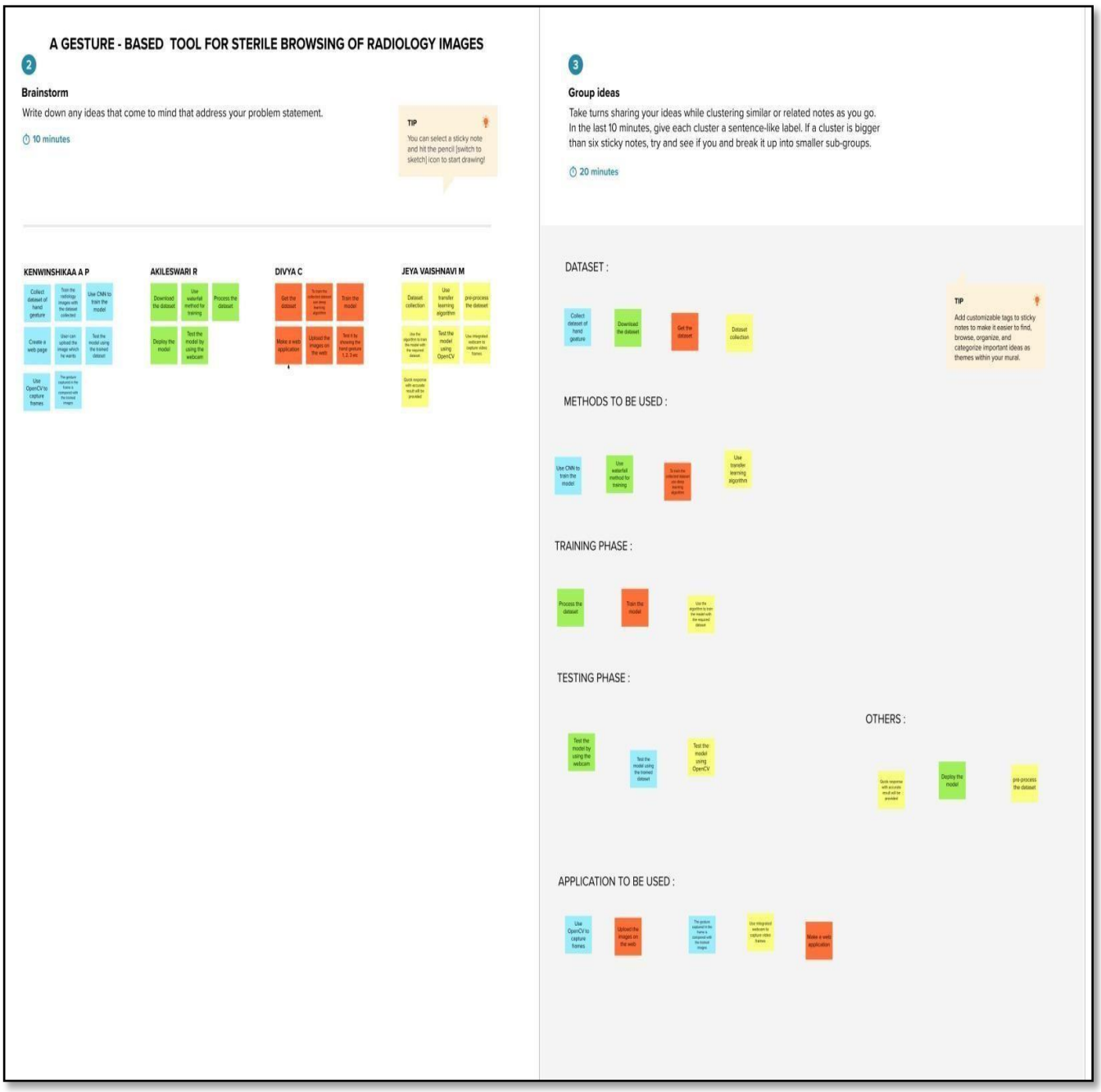
3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

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



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Interaction between doctor-computer inside the operation room. Keyboard and pointing devices such as mouse are today's method of human-computer interaction. However, the use of computer keyboard and mouse by doctors and nurses in intensive care units is a common method of spreading infections.
2.	Idea / Solution description	In this project, we use hand gestures as an alternative to existing interface techniques, offering the major advantage of sterility.
3.	Novelty / Uniqueness	We are using Convolutional Neural Network to first train the model on the images of different hand gestures, like showing numbers with fingers as 0,1,2,3,4,5. Then we made a web portal using Flask where user can input any image on which one wants to perform the operations. After uploading the image, our portal uses the integrated webcam to capture the video frame using OpenCV. The gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the prediction is 0 - then images is converted into rectangle, 1 - image is Resized into (200,200), 2 - image is rotated by -45°, 3 - image is blurred , 4 - image is Resized into (400,400) , 5 - image is converted into gray scale.
4.	Social Impact / Customer Satisfaction	Contributing the corporate social responsibility by providing better solutions to the healthcare and to patients.
5.	Business Model (Revenue Model)	Can collaborate with diagnosis centers and hospitals. It can also collaborate with government health awareness camps.
6.	Scalability of the Solution	The use of doctor-computer interaction devices in the operation room supports medical imaging manipulation while allowing doctors' hands to remain sterile, supporting their focus of attention, and providing fast response times.

3.4 Problem Solution fit

Define CS,

1. CUSTOMER SEGMENT(S) CS

Hospital domain, especially used for doctors insid the operation room.

6. CUSTOMER CONSTRAINTS CC

No breakdown of power and full internet access.

5. AVAILABLE SOLUTIONS AS

Doctors use their hands to access the computer to zoom, scroll, rotate etc of radiology images by touching keyboard, mouse and screen.

Explore AS,

Focus on J&P, tap into BE,

2. JOBS-TO-BE-DONE/PROBLEMS J&P

To keep the doctor sterile inside the operation roo

9. PROBLEM ROOT CAUSE RC

Doctors use their hands to access the computer by touching keyboard, mouse and screen to scroll, zoom, rotate the radiology images and this may lead to the spreading of infections inside the operation room.

7. BEHAVIOUR BE

Directly related : Easy to use, can predict the hand gesture correctly and accurately.

Indirectly associated : Require high internet speed.

Focus on J&P, tap into BE,

3. TRIGGERS TR

If the operation is done and successfully complet by using this project in an hospital, that makes oth hospital to use.

10. YOUR SOLUTION SL

1) Create a webpage to upload the images which

8. CHANNELS of BEHAVIOUR CH

Online: To upload the radiology images in the created webpage.

Identify strong TR & EM

4. EMOTIONS: BEFORE / AFTER EM

Before: Spreading of infection.

After: Faster response, avoid spreading of infection

needs in operation room.

2) It consumes less data and secures the information of the radiology images.

3) OpenCV is used to recognize the hand gestures that is used by the doctors.

4) Accurate result will be provided.

Offline: Stores the result of the radiology images.

Extract online & offline CH of BE

CHAPTER 4

REQUIREMENTANALYSS

4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Link (HTML page)
FR-2	User Confirmation	Confirmation via Email
FR-3	Upload images as input	Add image through Device or through Drive
FR-4	Camera on	When the camera is on it recognize the hand gestures and return the required output.

4.2 Non-functional Requirements

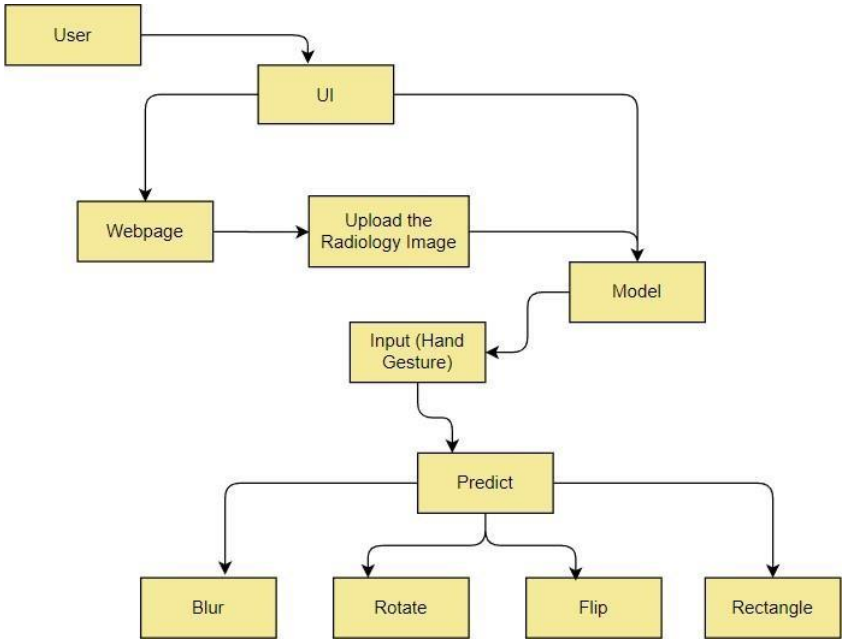
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	This software will be easy to use for all users with minimal instructions. 100% of the languages on the graphical user interface (GUI) shall be intuitive and understandable by non-technical users.
NFR-2	Security	The user of the system should be provided the surety that their account details are secure. The system will provide security against cross site request forgery.
NFR-3	Reliability	This software will be operable in all lighting conditions. Regardless of the brightness level in user's operating environment, the program shall always detect user's hands.

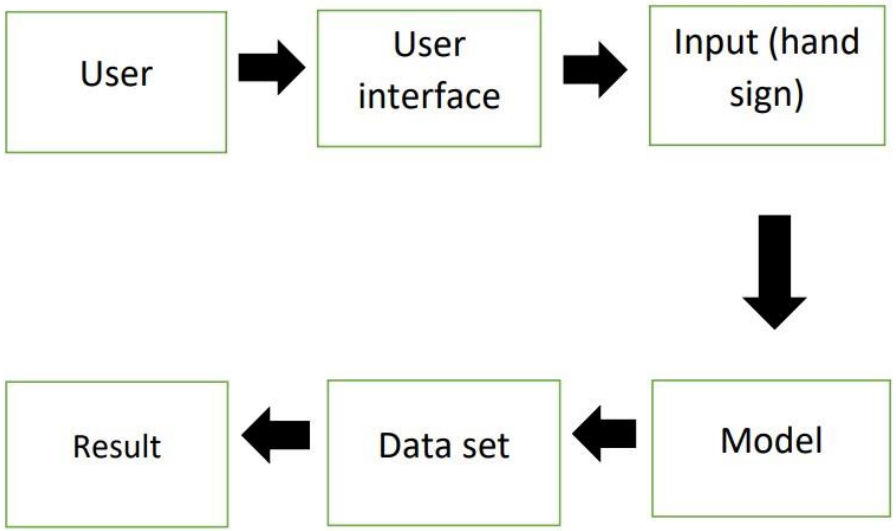
NFR-4	Performance	This software will minimize the number of calculations needed to perform image processing and hand gesture detection. Each captured video frame shall be processed fast.
NFR-5	Availability	This software will be available to all operating system. While it is currently has a relatively limited role in direct patient care, its evolving role in complex clinical decision making is foreseeable.
NFR-6	Scalability	This software will be enterprise scalability of AI development and deployment.

CHAPTER 5

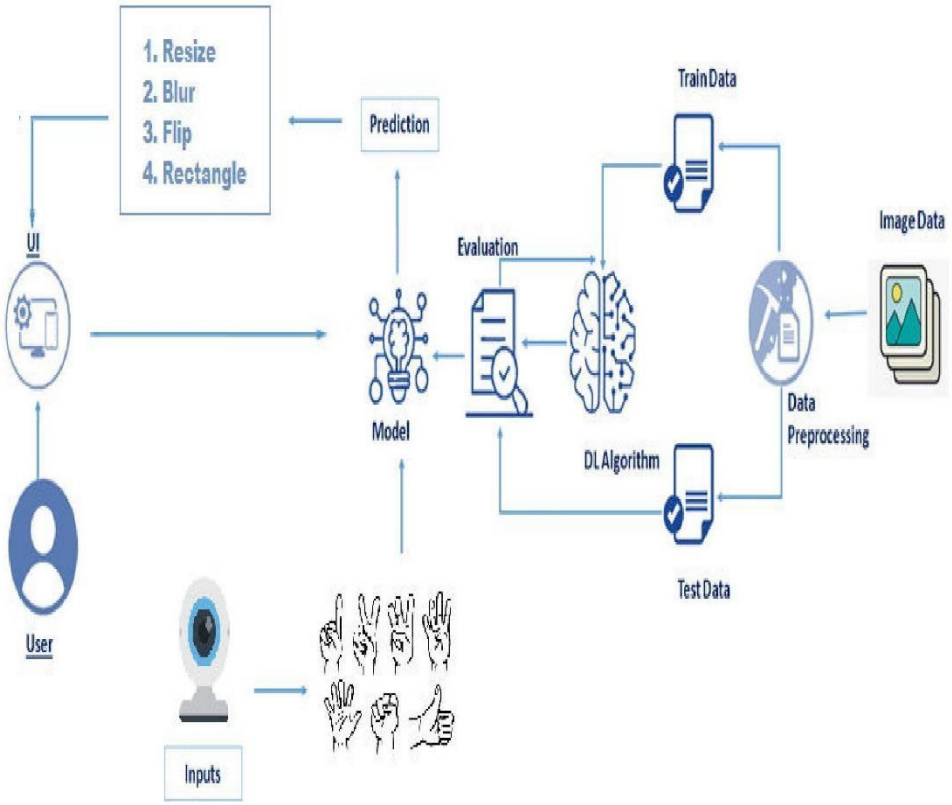
PROJECT DESIGN

5.1 Data flow diagams





5.2 Solution Architecture



Technical Architecture

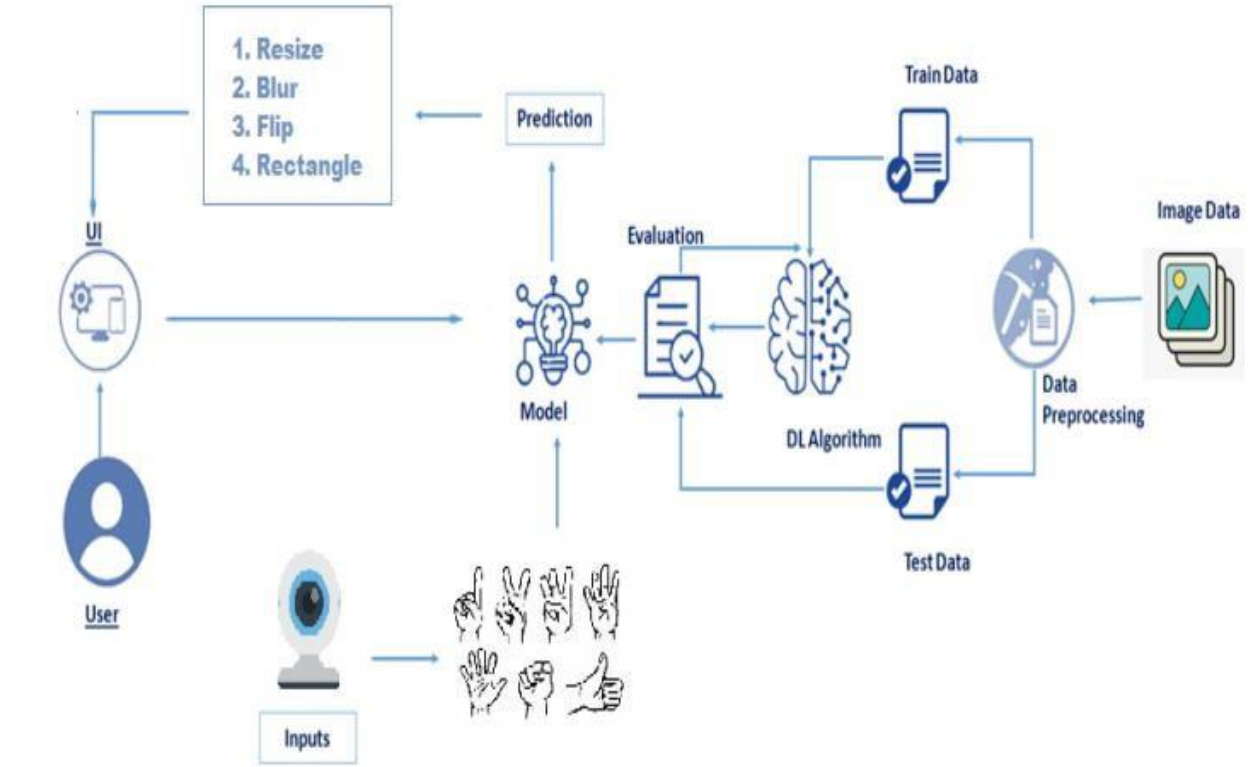


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
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:

S.No	Characteristics	Description	Technology
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

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:

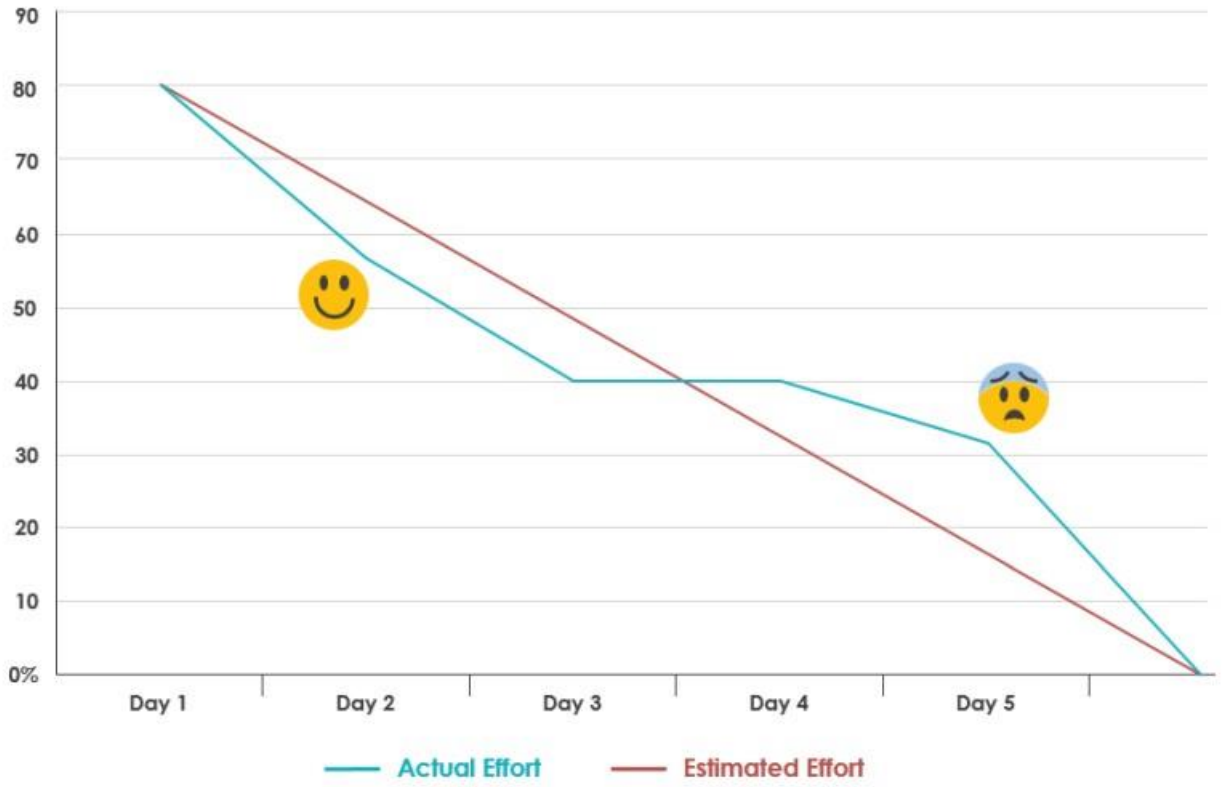
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$Team\ Velocity = \Sigma \text{ Sprint } 1 + \text{Sprint } 2 + \cdots = 20 + 15 + 10 + 5\ 4 = 12.5$$

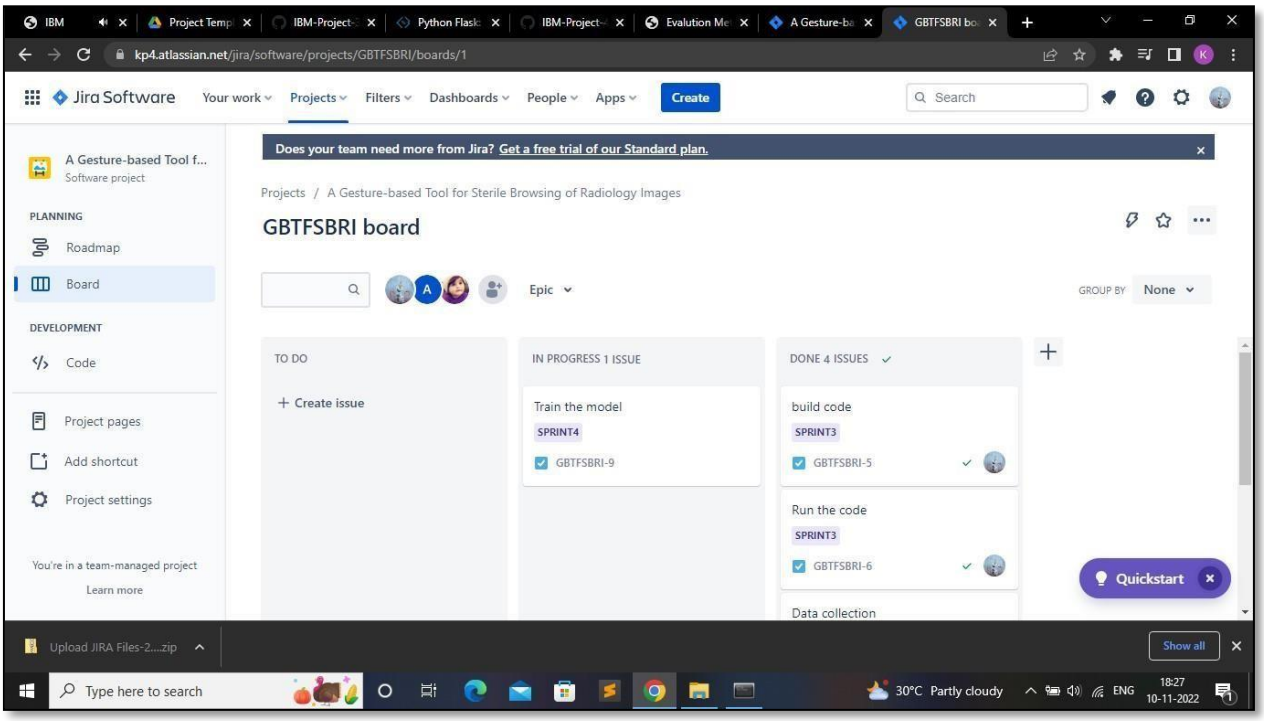
$$Total\ Sprint\ AV = Team\ Velovity\ Duration = 12.5\ 6 = 2.08$$

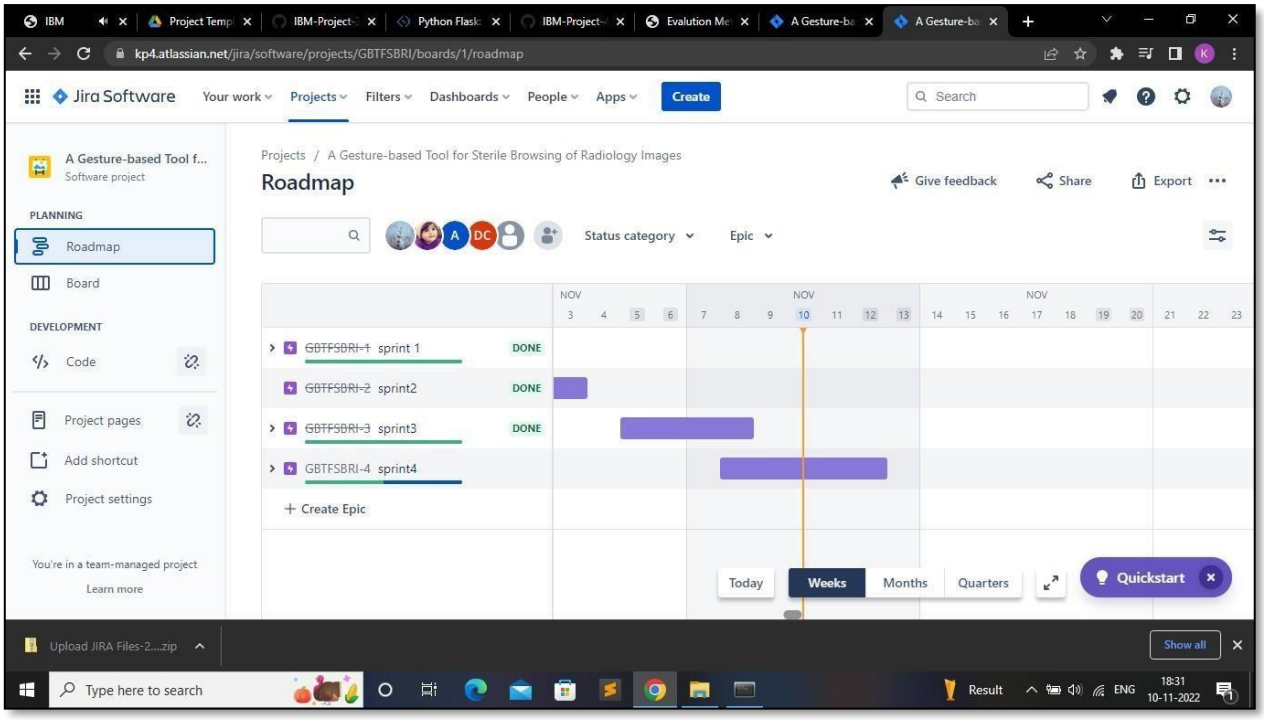
Burn down Chart:

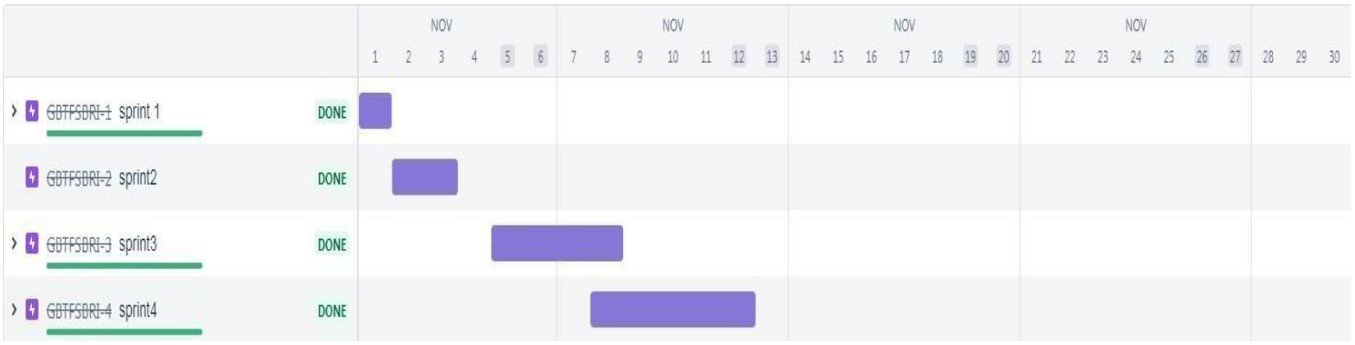
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.


90
80
70
60
50
40
30
20
10
0%
Day 1
Day 2
Day 3
Day 4
Day 5
Actual Effort
Estimated Effort

6.3 Reports from JIRA






NOV
1 2 3 4 5 6
NOV
7 8 9 10 11 12 13
NOV
14 15 16 17 18 19 20
NOV
21 22 23 24 25 26 27
28 29 30
GBTFSDR1-1 sprint 1
DONE
GBTFSDR1-2 sprint2
DONE
GBTFSDR1-3 sprint3
DONE
GBTFSDR1-4 sprint4
DONE

CHAPTER 7

CODING & SOLUTIONING

main.css

```
.img-preview {     width:
256px;     height: 256px;
position: relative;     border:
5px solid #F8F8F8;
   box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);     margin-top:
1em;     margin-bottom: 1em;
}

.img-preview>div {     width:
100%;     height: 100%;
background-size: cover;
background-repeat: no-repeat;
background-position: center;
}

input[type="file"] {
display: none;
}

.upload-label{     display:
inline-block;     padding:
12px 30px;
background: #39D2B4;
color: #fff;     font-size:
1em;     transition: all
.4s;
cursor: pointer;
}

.upload-label:hover{
background: #34495E;     color:
#39D2B4;
}

.loader {     border: 8px solid #f3f3f3; /*
Light grey */     border-top: 8px solid
```

```
#3498db; /* Blue */     border-radius: 50%;
   width: 50px;
height: 50px;
   animation: spin 1s linear infinite;
}

@keyframes spin {
   0% { transform: rotate(0deg); }
   100% { transform: rotate(360deg); }
}
```

main.js

```
$(document).ready(function () {
   // Init
   $('.image-section').hide();
   $('.loader').hide();
   $('#result').hide();

   // Upload Preview     function
readURL(input) {        if (input.files &&
input.files[0]) {            var reader =
new FileReader();
reader.onload = function (e) {
            $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
            $('#imagePreview').hide();
            $('#imagePreview').fadeIn(650);
         }
         reader.readAsDataURL(input.files[0]);
      }
   }
   $("#imageUpload").change(function () {
      $('.image-section').show();
      $('#btn-predict').show();
      $('#result').text('');
$('#result').hide();
      readURL(this);
   });

   // Predict
   $('#btn-predict').click(function () {        var form_data =
```

```
new FormData($('#upload-file')[0]);

        // Show loading animation
        $(this).hide();
       $('.loader').show();
        // Make prediction by calling api /predict
        $.ajax({            type: 'POST',
url: '/predict',            data: form_data,
contentType: false,            cache: false,
processData: false,            async: true,
success: function (data) {                // Get
and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
$('#result').html(data);                console.log('Success!');
            },
        });
    });

});


home.html

<html>
<script>


</script>


<style>
.header { position: relative; top:0;
margin:0px; z-index: 1; left: 0px; right: 0px;
position: fixed; background-color:FFFFFF ;
color: rgb(181, 228, 236); box-shadow: 0px
8px 4px rgb(10, 102, 109); overflow:
hidden; padding-left:20px; font-family:
'Times New Roman'; font-size: 2vw;
width:          100%;
height:8%;   textalign:
        center;            }
.topnav        {
```

```
overflow: hidden;
  background-color: #87CEFA;
}

.topnav-right a {   float: left;   color: black;
text-align: center;   padding: 14px 16px;
text-decoration: none;   font-size: 18px; }
.topnav-right a.active {   background-color:
#07201e;   color: rgb(238, 226, 234); }
.topnav-right a:hover {   background-color:
rgb(181,
228, 236);   color: rgb(6, 27, 36);
}

.topnav-right {   float: right;

       padding-
right:100px;
}

body {

  background-color: rgb(88, 129, 123) ;   background-
repeat: no-repeat;   background-size:cover;
  background-position: 0px 0px;
}   .button {   backgroundcolor:
#091425;          border:    none;
color:    rgb(181,    228,    236);
padding:  15px  32px;       text-
align: center;    textdecoration:
none;      display:   inline-block;
font-size: 12px;
  border-radius: 16px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {
width: 100%;   padding: 12px 20px;
display: inline-block;   marginbottom:18px;
border: 1px solid #ccc;
```

```
box-sizing: border-box;
}

button {    background-color:
#091425;   color: rgb(181,
228, 236);   padding: 14px
20px;   marginbottom:10px;
border: none;   cursor:
pointer;   width: 17%;
border-radius:4px;
fontfamily:Montserrat;
}

button:hover {
opacity: 0.8;
}

.cancelbtn {   width: auto;   padding:
10px 18px;    background-color:
rgb(181, 228, 236);
}

.imgcontainer {   textalign:
center;   margin:
24px 0 12px 0;
}

img.avatar {  width:
30%;
borderradius: 50%;
}

.container {
padding: 16px;
}

span.psw {    float:
right;   paddingtop:
16px;
}

/* Change styles for span and cancel button on extra small screens */
```

```
@media screen and (max-width: 300px) {
span.psw {       display: block;       float:
none;   }   .cancelbtn {
width: 100%;
  }
}

.home{ margin:80px;

  width: 84%;
height: 500px;   padding-
top:10px;
padding-left: 30px;

}
.login{ margin:80px; box-sizing:
content-box;     width: 84%;
height: 420px;   padding: 30px;
border: 10px solid rgb(13, 53, 68);
}
.left,.right{     box-sizing:   content-
box;  height: 400px;  margin:20px;
border: 10px solid rgb(13, 53, 68);
}
.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {   max-
width: 1000px;   position:
relative;   margin:
auto;
}

/* Caption text */
.text {   color:
#9ac0c0;   font-size:
15px;   padding: 8px
12px;   position:
absolute;   bottom:
8px;   width: 100%;
text-align: center;
}
```

```
/* The dots/bullets/indicators */
.dot {  height: 15px;  width: 15px;
margin: 0 2px;  background-color:
#bbb;  border-radius: 50%;  display:
inline-block;  transition:
backgroundcolor 0.6s ease;
}

.active {  color: rgb(145,
216, 221);
}

/* Fading animation */
.fade {
  -webkit-animation-name: fade;  webkit-
animation-duration:          1.5s;
animation-name: fade;
animationduration: 1.5s;
}

@-webkit-keyframes fade {  from
{opacity: .4}
to {opacity: 1}
}

@keyframes fade {
from {opacity: .4}  to
{opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}
```

```
@import url("https://fonts.googleapis.com/css?family=Luckiest+Guy");
/*  BODY  */ body  {
position:  absolute;
top:  0;      left:  0;
right: 0;  bottom: 0;
```

```
width:          100%;
height:         100%;
overflow: hidden;
  font-family: "Arial", cursive;   -webkit-font-smoothing:
antialiased;
}

::selection {
  background: transparent;
}
/*     CLOUDS     */
body:before        {
content:           "";
position: absolute;
top:  0;      left:  0;
right: 0;   width: 0;
height: 0;   margin:
auto;
 border-radius: 100%;  background-
  image:
  url("https://cdn.dribbble.com/users/237905/screenshots/2632750/handanimation.gif");   display:
  block;   box-shadow: 0 0 150px 100px rgba(255, 255, 255, 0.6),
   200px 0 200px 150px rgba(255, 255, 255, 0.6),
   -250px 0 300px 150px rgba(255, 255, 255, 0.6),
550px 0 300px 200px rgba(255, 255, 255, 0.6),
   -550px 0 300px 200px rgba(255, 255, 255, 0.6);
}
/* JUMP */ h1 {   cursor:
       default;
position: absolute;
top: 0;   left: 0;
right: 0;   bottom:
0;   width: 100%;
height: 100px;
margin: 70px;
display: block;
text-align: center;
}

h1 span {    position:
relative;     top:  5px;
display: inline-block;
```

```
font-size:         25px;
color: #061a1f;
  text-shadow: 0 1px 0 rgb(151, 201, 197), 0 2px 0 rgb(151, 201, 197), 0 3px 0 rgb(151, 201,
197), 0 4px 0 rgb(151, 201, 197),
    0 5px 0 rgb(151, 201, 197), 0 6px 0 transparent, 0 7px 0 transparent, 0 8px 0 transparent,
0 9px 0 transparent, 0 10px 10px rgba(58, 159, 167, 0.4); }

h1 span:nth-child(2) {
  -webkit-animation-delay: 0.1s;
}

h1 span:nth-child(3) {
  -webkit-animation-delay: 0.2s;
}

h1 span:nth-child(4) {
  -webkit-animation-delay: 0.3s;
}
h1 span:nth-child(5) {
  -webkit-animation-delay: 0.4s;
}

h1 span:nth-child(6) {
  -webkit-animation-delay: 0.5s;
}

h1 span:nth-child(7) {
  -webkit-animation-delay: 0.6s;
}

h1 span:nth-child(8) {
  -webkit-animation-delay: 0.2s;
}

h1 span:nth-child(9) {
  -webkit-animation-delay: 0.3s;
}

h1 span:nth-child(10) {   -webkit-animation-
delay: 0.4s;
}

h1 span:nth-child(11) {   -webkit-animation-
```

```
delay: 0.5s;
}

h1 span:nth-child(12) {   -webkit-animation-
delay: 0.6s;
}

h1 span:nth-child(13) {   -webkit-animation-
delay: 0.7s;
}

h1 span:nth-child(14) {   -webkit-animation-
delay: 0.8s;
}

/* ANIMATION */
@-webkit-keyframes bounce {
  100% {
 top: -20px;
    text-shadow: 0 1px 0 #ccc, 0 2px 0 #ccc, 0 3px 0 #ccc, 0 4px 0 #ccc,      0 5px 0 #ccc, 0 6px 0 #ccc, 0
  7px 0 #ccc, 0 8px 0 #ccc, 0 9px 0 #ccc,
     0 50px 25px rgba(0, 0, 0, 0.2);
  }
}

</style>

<body>
<h1 style="color: rgb(193, 207, 207);">
  <table style="width:100%">

  </table>
   <br>
   <span>HAND   GESTURE   RECOGNITION</span>
  </br>
    </h1>
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:#c4dfd7; paddingtop:1%;padding-
left:5%;"><b><h1>Hand Gesture</b></h1></h1></div>   <div
class="topnav-right"style="padding-top:0.5%;color:white">
   <a  class="active" href="{{ url_for('home')}}"><u>Home</u></a>
   <a  class="active" href="{{ url_for('intro')}}">Introduction</a>
   <a  class="active" href="{{ url_for('image1')}}">Launch</a>
```

```
  </div>
</div>

</body>

</html>
```

intro.html

```
<html>
<script>


</script>


<style>
.header { position: relative;
top:0; margin:0px; z-index:
1;  left: 0px; right: 0px; position: fixed;
background-color: rgb(10, 102, 109) ;
color: white; box-shadow: 0px
8px 4px grey; overflow: hidden;
padding-left:20px; font-family:
'Josefin Sans'; font-size: 2vw;
width: 100%; height:8%;
textalign: center; } .topnav {
overflow: hidden;
  background-color: #FCAD98;
}

.topnav-right a {   float:
left;        color:   black;
text-align:        center;
padding:   14px   16px;
text-decoration: none;
font-size: 18px;
}

.topnav-right          a.active          {
background-color: #07201e;   color:
rgb(238, 226, 234);
```

```
}

.topnav-right a:hover {   backgroundcolor:
rgb(181, 228, 236);   color: rgb(6,
27, 36);
}

.topnav-right {   float: right;

      padding-
right:100px;
}

body { background-

  color: ;  backgroundrepeat:

      no-repeat;

  background-

  size:cover;   background-

  image: url("https://i.pinimg.

  com/originals/b2/1d

  /c6/b21dc69346915

  015bc4e19bd502f40

  1b.gif");
  background-size:

  cover;

  backgroundposition:

  0px 0px;

  }
  .button {    background-
color: #091425;    border:
none;   color: white;
```

```
padding: 15px 32px;   text-
align: center;
textdecoration: none;
display: inline-block;   font-
size: 12px;
border-radius: 16px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {
width: 100%;   padding: 12px 20px;
display: inline-block;   marginbottom:18px;
border: 1px solid #ccc;   box-sizing: border-
box;
}

button {    background-color:
#091425;         color:   white;
padding:       14px       20px;
marginbottom:10px;
border:   none;         cursor:
pointer;          width:    17%;
borderradius:4px;
fontfamily:Montserrat;
}

button:hover { opacity:
  0.8;
}

.cancelbtn {   width: auto;   padding:
       10px    18px;
background-color: #f44336;
}

.imgcontainer {   textalign:
center;   margin:
24px 0 12px 0;
}
```

```
img.avatar {   width:
30%;   border-radius:
50%;
}

.container {
padding: 16px;
}

span.psw {    float:
right;   padding-
top: 16px;
}
/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
span.psw {        display: block;        float:
none;   }    .cancelbtn {
    width: 100%;
  }
}

.home{ margin:80px;

  width: 84%;
height: 500px;   padding-
top:10px;
padding-left: 30px;
}
.login{ margin:80px;
boxsizing: content-box;
width: 84%;   height:
420px;   padding: 30px;
border: 10px solid blue;
}
.left,.right{  box-sizing: content-
box;             height: 400px;
                 margin:20px;
border: 10px solid blue;
}

.mySlides {display: none;} img
{vertical-align: middle;}
```

```
/* Slideshow container */
.slideshow-container {   max-
width: 1000px;   position:
relative;   margin:
auto;
}

/* Caption text */
.text {   color:
#f2f2f2;   font-size:
15px;   padding: 8px
12px;   position:
absolute;   bottom:
8px;   width: 100%;
text-align: center;
}
/* The dots/bullets/indicators
*/  .dot {   height: 15px;   width:
15px;   margin: 0 2px;
background-color: #bbb;
border-radius: 50%;   display:
inline-
block;    transition: background-color
0.6s ease;
}
.active {
  background-color: #FCAD98;
}

/* Fading animation */
.fade {
  -webkit-animation-name: fade;   webkit-
animation-duration:           1.5s;
animation-name: fade;
animationduration: 1.5s;
}

@-webkit-keyframes fade {
from {opacity: .4}   to
{opacity: 1}
}
```

```
@keyframes fade {
from {opacity: .4}   to
{opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}



@import url("https://fonts.googleapis.com/css?family=Montserrat&display=swap");

* {
padding: 0;
margin: 0;
}

body {   height: 100vh;   display:
flex;   flexdirection: column;
justify-content: center;   align-
items: center;
}

h1 {
 font-family: "Montserrat Medium";
  max-width: 90ch;   text-
align: center;   transform:
scale(0.94);
  animation: scale 3s forwards cubic-bezier(0.5, 1, 0.89, 1);
}
@keyframes scale {   100%
{
    transform: scale(1);
  }
}

span {   display: inlineblock;
opacity: 0;   filter: blur(4px);
}
```

```
span:nth-child(1) {    animation: fade-in 1s 0.1s forwards cubic-
bezier(0.11, 0, 0.5, 0);
}

span:nth-child(2) {    animation: fade-in 0.8s 0.2s forwards cubic-
bezier(0.11, 0, 0.5, 0);
}

span:nth-child(3) {    animation: fade-in 0.8s 0.3s forwards cubic-
bezier(0.11, 0, 0.5, 0);
}

span:nth-child(4) {    animation: fade-in 0.8s 0.4s forwards cubic-
bezier(0.11, 0, 0.5, 0);
}

span:nth-child(5) {    animation: fade-in 0.8s 0.5s forwards cubic-
bezier(0.11, 0, 0.5, 0);
}

span:nth-child(6) {    animation: fade-in 0.8s 0.6s forwards cubic-
bezier(0.11, 0, 0.5, 0); }

span:nth-child(7) {    animation: fade-in 0.8s 0.7s forwards cubic-
bezier(0.11, 0, 0.5, 0); }  span:nth-child(8) {
  animation: fade-in 0.8s 0.8s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(9) {    animation: fade-in 0.8s 0.9s forwards cubic-
bezier(0.11, 0, 0.5, 0);
}

span:nth-child(10) {
  animation: fade-in 0.8s 1s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(11) {
  animation: fade-in 0.8s 1.1s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(12) {
  animation: fade-in 0.8s 1.2s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
```

```
span:nth-child(13) {
  animation: fade-in 0.8s 1.3s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(14) {
  animation: fade-in 0.8s 1.4s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(15) {
  animation: fade-in 0.8s 1.5s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(16) {
  animation: fade-in 0.8s 1.6s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(17) {
  animation: fade-in 0.8s 1.7s forwards cubic-bezier(0.11, 0, 0.5, 0);
}

span:nth-child(18) {   animation: fade-in 0.8s 1.8s forwards cubic-
bezier(0.11, 0, 0.5, 0);
} span:nth-child(19)
{
  animation: fade-in 0.8s 1.9s forwards cubic-bezier(0.11, 0, 0.5, 0); }
span:nth-child(20) {
  animation: fade-in 0.8s 2.0s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
span:nth-child(21) {
  animation: fade-in 0.8s 2.1s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
span:nth-child(22) {
  animation: fade-in 0.8s 2.2s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
span:nth-child(23) {
  animation: fade-in 0.8s 2.3s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(24) {
  animation: fade-in 0.8s 2.4s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(25) {
  animation: fade-in 0.8s 2.5s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(26) {
  animation: fade-in 0.8s 2.6s forwards cubic-bezier(0.11, 0, 0.5, 0);
```

```
}span:nth-child(27) {
  animation: fade-in 0.8s 2.7s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(28) {
  animation: fade-in 0.8s 2.8s forwards cubic-bezier(0.11, 0, 0.5, 0); }
@keyframes fade-in {
  100% {
opacity: 1;
filter: blur(0);
  }
}

</style>

<body>
  <h1>INTRODUCTION</h1>
<h1>
  <span> Hand Gesture </span> <span> recognition system </span> <span> provides us </span >
<span> an innovative,</span> <span>natural,</span> <span>  user friendly </span> <span> way
of interaction </span > <span> with the computer</span>
  <span> which is more </span> <span> familiar to the </span> <span>human beings. </span >
<span> In our project, </span> <span> the hand region </span> <span> is extracted from
</span> <span> the background  </span > <span> by using</span>
<span> Region of intrest. </span> <span> Then, </span> <span> we will be </span >
<span>predicting the labels </span> <span> based on the </span> <span> CNN trained model weights
</span> <span> of hand gestures </span > <span> using that predicted labels</span>
<span> we apply if conditions </span> <span> to control some of the actions
</span>  <span>like </span > <span>reshaping , blur, flip of the given
image.</span>  </h1>
<!--Brian Tracy-->

<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:#c1e2d9;
paddingtop:1%;paddingleft:5%;">Hand Gesture System</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a class="active" href="{{ url_for('intro')}}"><u>Introduction</u></a>
    <a class="active" href="{{ url_for('image1')}}">Launch</a>
  </div>
</div>
</body>

</html>
```

launch.html

```
<html lang="en">

 <head>
 <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=0.6">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
<meta charset="UTF-8">
    <title>Predict</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">

    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
<style>
 .bar
 {
 margin: 0px;
padding:20px;   backgroundcolor:black;
opacity:0.6; color:black;  font-
family:'Roboto',sans-serif; font-style:
italic; border-radius:20px; font-size:25px;
  } a { color:black;
  float:right;
  textdecoration:none;
 font-style:normal;
paddingright:20px;    }
a:hover{
background-color:black;
color:black;   fontsize:30px;
paddingleft:10px;
 }

 div1{    text-align:
center;    width:
650spx;    height:
```

```
800px;    padding:
190px;    margin:
10px;    position:
absolute;
 }



 body
 {
    background-image: url("https://img3.goodfon.com/wallpaper/nbig/a/af/ruki-znakisteny.jpg");
background-size: cover;
 }

 .header { position: relative;
      top:0;
margin:0px;       zindex:
1;       left:
0px;       right: 0px;
 position: fixed;
      background-color: rgb(10, 102, 109) ;
 color: black;
      box-shadow: 0px 8px 4px
 grey;       overflow: hidden;
 padding-left:20px;      fontfamily:
 'Josefin Sans';      fontsize: 2vw;
 width: 100%;      height:8%;
 text-align:
 center;
 }
    .topnav {
overflow: hidden;
   background-color: #056959;
 }

 .topnav-right a {
float: left;    color:
black;    text-align:
center;    padding: 14px
16px;    textdecoration:
none;
font-size: 18px;
```

```
 }

 .topnav-right a.active {     backgroundcolor:
#07201e;
   color: rgb(238, 226, 234);
 }

 .topnav-right a:hover {     backgroundcolor:
rgb(181, 228, 236);     color: rgb(6,
27, 36);
 }



 .topnav-right {
float: right;
padding-right:100px;
 }


 .button {   background-color:
#091425;
  border: none; color:
  black; padding: 15px
  32px; text-align:
  center; textdecoration:
  none; display: inline-
  block; font-size: 12px;
  border-radius: 16px;
  }
  .button:hover {
    box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
  }   form {border: 2px solid black; margin-
left:400px;marginright:400px;}

 input[type=text], input[type=password] {
width: 100%;     padding: 12px 20px;
display: inline-block;     marginbottom:18px;
border: 1px solid #ccc;      box-sizing:
border-box;
 }
```

```
  button {
    background-color: #091425;
    color: black;
padding: 14px 20px;
margin-bottom:10px;
border: none;     cursor:
pointer;     width: 17%;
border-radius:4px;
    font-family:Montserrat;
  }

  button:hover {
opacity: 0.8;
  }

  .cancelbtn {     width: auto;
padding: 10px
18px;      background-color:
#f44336;
  }

  .imgcontainer {   textalign:
  center;   margin:
  24px 0 12px 0;
  }
  img.avatar {   width:
  30%;   border-radius:
  50%; }

  .container {
  padding: 16px;
  }

  span.psw {      float:
right;     paddingtop:
16px;
  }
  /* Change styles for span and cancel button on extra small screens */
  @media screen and (max-width: 300px) {
    span.psw {
display: block;
float: none;     }
```

```
.cancelbtn {       width:
100%;
   }
 }

 .home{
margin:80px;

   width: 84%;
height: 500px;
padding-top:10px;
padding-left: 30px;

 }  .login{    margin:80px;    box-
sizing: content-box;      width: 84%;
height: 420px;    padding: 30px;
border: 10px solid rgb(12, 91, 94);
} .left,.right{  box-sizing: content-
box;  height: 400px;  margin:20px;
border: 10px solid rgb(12, 91, 94); }

 .mySlides {display: none;} img {vertical-
 align: middle;}

 /* Slideshow container */
 .slideshow-container {
max-width: 1000px;
position: relative;    margin:
auto;
}

 /* Caption text */
.text {    color: #f2f2f2;
font-size:    15px;
padding: 8px 12px;
position:      absolute;
bottom: 8px;    width:
100%;    text-align:
center;
 }
 /* The dots/bullets/indicators */
.dot {    height: 15px;    width: 15px;
margin: 0 2px;    background-color:
```

```
#bbb;     border-radius: 50%;
display: inline-block;     transition:
backgroundcolor 0.6s ease;
 }

 .active {     background-color:
#267481;
 }

 /* Fading animation */
 .fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration:
 1.5s;         animation-name:    fade;
 animationduration: 1.5s;
 }
 @-webkit-keyframes fade {
 from {opacity: .4}   to
 {opacity: 1} }

 @keyframes fade {
 from {opacity: .4}   to
 {opacity: 1}
 }
 /* On smaller screens, decrease text size */
 @media only screen and (max-width: 300px) {
  .text {font-size: 11px}
 }

 </style>
 </head>

 <body>
 <div class="header">
 <div           style="width:50%;float:left;font-size:2vw;text-align:left;color:#c1e2d9;
paddingtop:1%;padding-left:5%;">Hand Gesture System</div>     <div
class="topnavright"style="padding-top:0.5%;">

    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a class="active" href="{{ url_for('intro')}}">Introduction</a>
    <a class="active" href="{{ url_for('image1')}}"><u>Launch</u></a>     </div>
</div>
 <br>
```

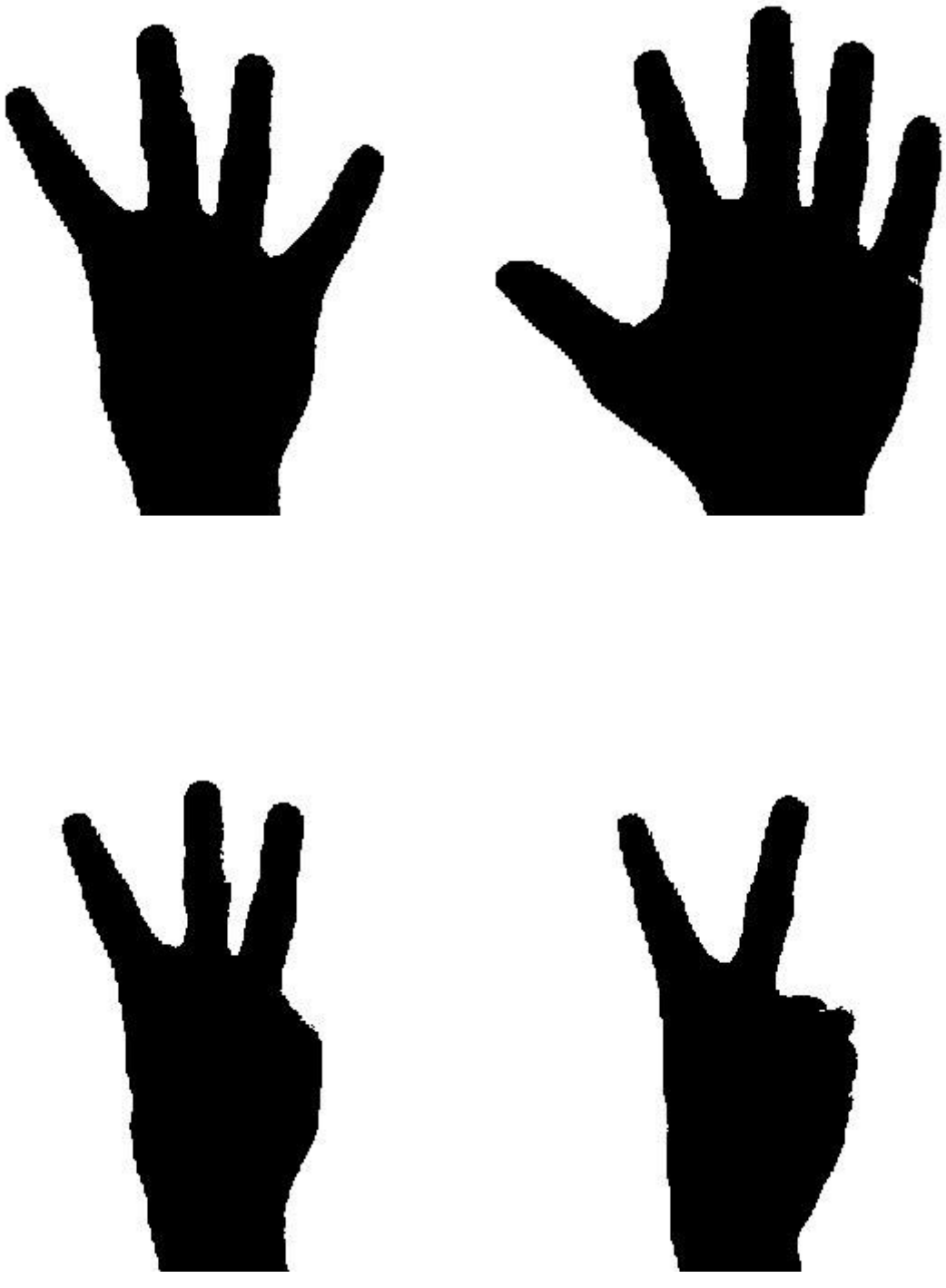
```
   <div1 style=""><h1><font color="Black" size="6" font-family="Roboto">Hand
Gesture</h1><br>
   <p><i><font color="Black" size="4" fonr-family="sans-serif"></i>Provide an image for which you
want to perform various operations</p>
   <br>
        <div>
         <h4>Upload Image Here</h4>
       <form action = "http://localhost:5000/" id="upload-file" method="post"
enctype="multipart/form-data">
        <label for="imageUpload" class="upload-label">
         Choose...
  </label>
        <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg,.pdf">
       </form>
           <center>
       <div class="image-section" style="display:none;">
        <div class="img-preview">
         <div id="imagePreview">
         </div>
        </div>
  <div>
         <button type="button" class="btn btn-info btn-lg " id="btn-predict">Predict!</button>
        </div>
      </div>
      <div class="loader" style="display:none;"></div>
           </center>
    </div>
    </div1>


        <footer>
    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>

  </html>
```

CHAPTER 8

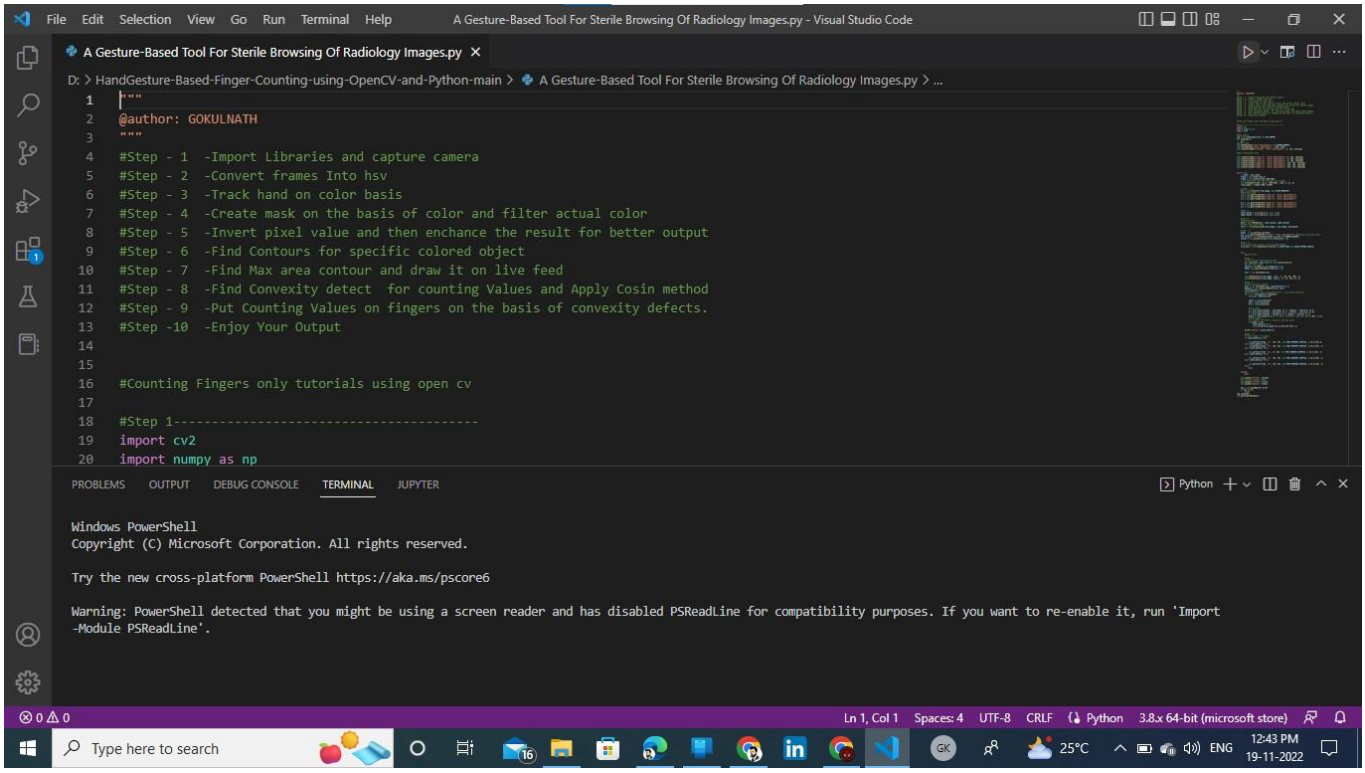
TESTING

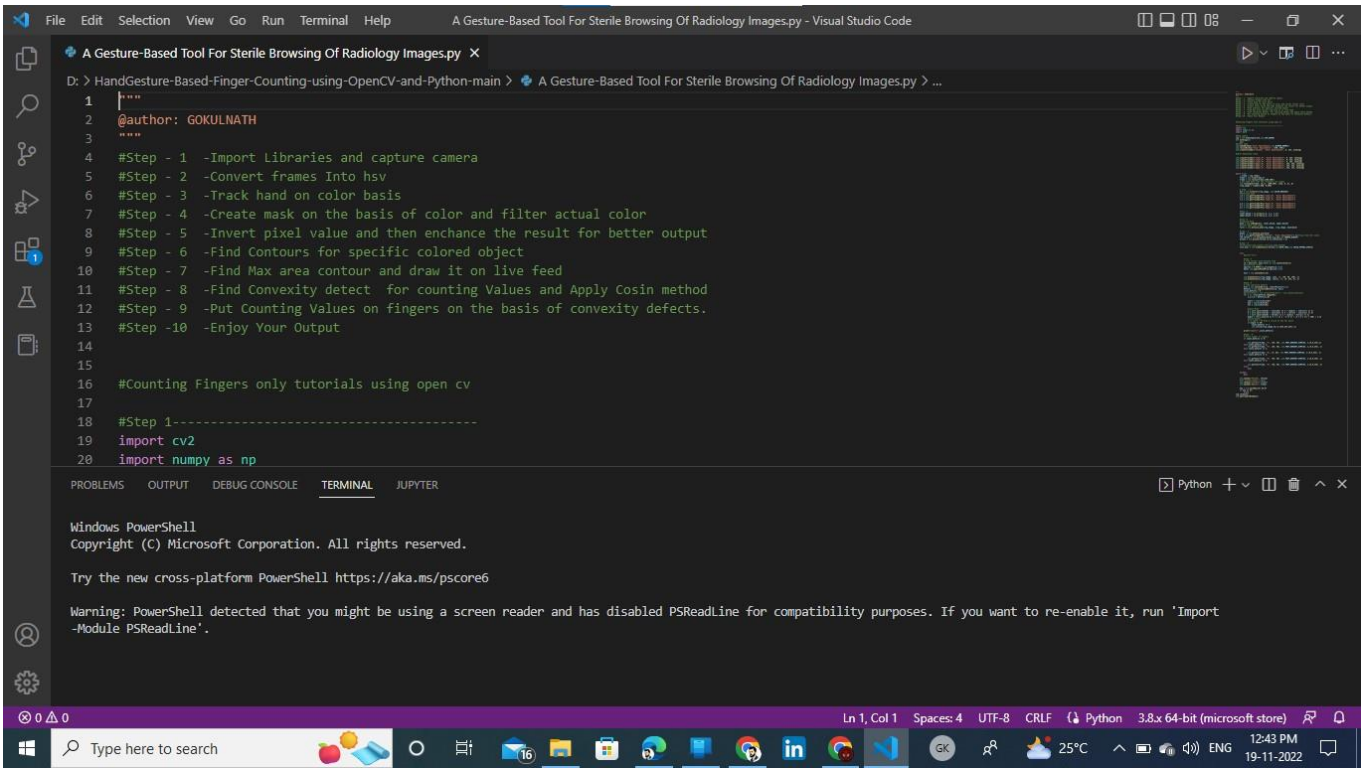


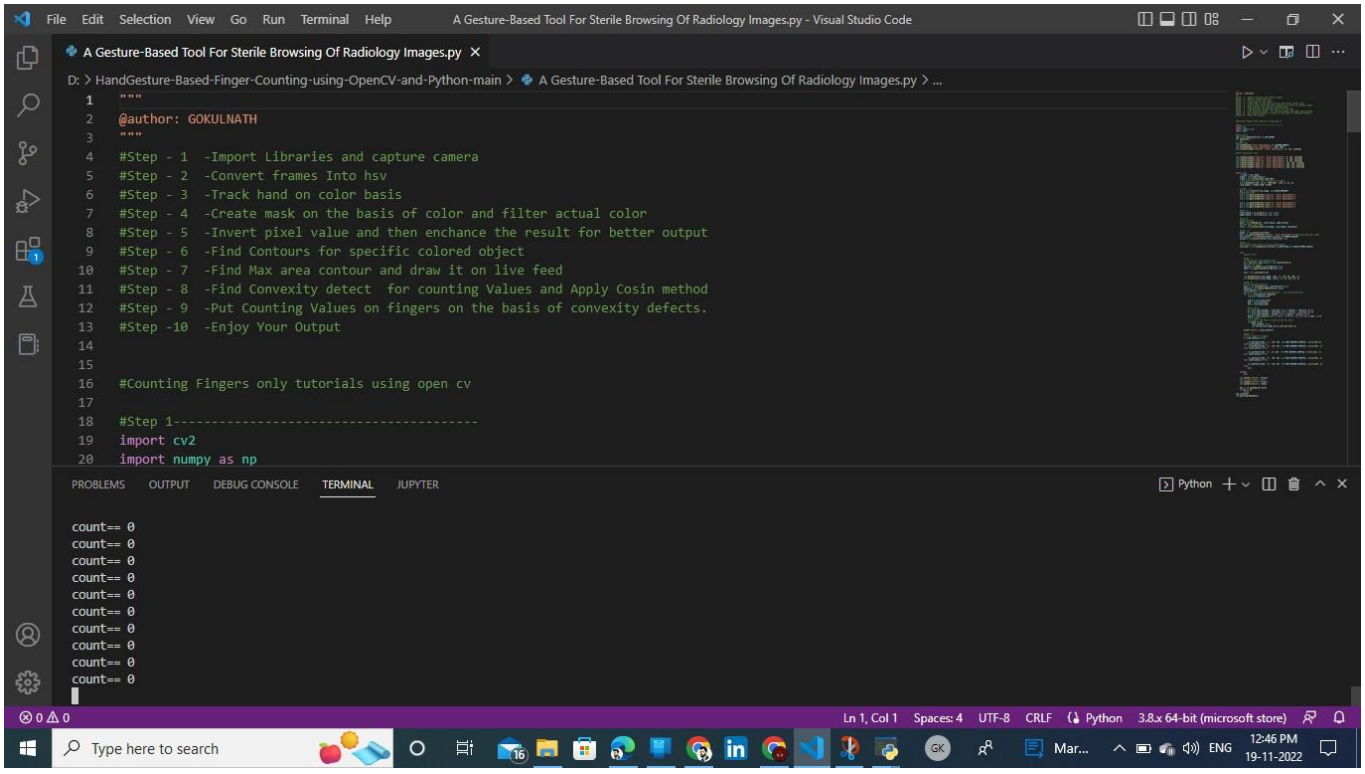
CHAPTER 9
RESULTS

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.




File Edit Selection View Go Run Terminal Help
A Gesture-Based Tool For Sterile Browsing Of Radiology Images.py - Visual Studio Code
A Gesture-Based Tool For Sterile Browsing Of Radiology Images.py
D: > HandGesture-Based-Finger-Counting-using-OpenCV-and-Python-main > A Gesture-Based Tool For Sterile Browsing Of Radiology Images.py > ...
"""
@author: GOKULNATH
"""
#Step - 1 -Import Libraries and capture camera
#Step - 2 -Convert frames Into hsv
#Step - 3 -Track hand on color basis
#Step - 4 -Create mask on the basis of color and filter actual color
#Step - 5 -Invert pixel value and then enchance the result for better output
#Step - 6 -Find Contours for specific colored object
#Step - 7 -Find Max area contour and draw it on live feed
#Step - 8 -Find Convexity detect for counting Values and Apply Cosin method
#Step - 9 -Put Counting Values on fingers on the basis of convexity defects.
#Step -10 -Enjoy Your Output
#Counting Fingers only tutorials using open cv
#Step 1----------
import cv2
import numpy as np
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import -Module PSReadLine'.
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.8.x 64-bit (microsoft store)
Type here to search
25°C
ENG
12:43 PM
19-11-2022


File Edit Selection View Go Run Terminal Help
A Gesture-Based Tool For Sterile Browsing Of Radiology Images.py - Visual Studio Code
A Gesture-Based Tool For Sterile Browsing Of Radiology Images.py
D: > HandGesture-Based-Finger-Counting-using-OpenCV-and-Python-main > A Gesture-Based Tool For Sterile Browsing Of Radiology Images.py > ...
"""
@author: GOKULNATH
"""
#Step - 1 -Import Libraries and capture camera
#Step - 2 -Convert frames Into hsv
#Step - 3 -Track hand on color basis
#Step - 4 -Create mask on the basis of color and filter actual color
#Step - 5 -Invert pixel value and then enchance the result for better output
#Step - 6 -Find Contours for specific colored object
#Step - 7 -Find Max area contour and draw it on live feed
#Step - 8 -Find Convexity detect for counting Values and Apply Cosin method
#Step - 9 -Put Counting Values on fingers on the basis of convexity defects.
#Step -10 -Enjoy Your Output
#Counting Fingers only tutorials using open cv
#Step 1----------
import cv2
import numpy as np
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Python
count== 0
count== 0
count== 0
count== 0
count== 0
count== 0
count== 0
count== 0
count== 0
count== 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.8.x 64-bit (microsoft store)
Type here to search
Mar...
ENG
12:46 PM
19-11-2022

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 t o 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 cap ture 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

app.py

```
"""

@author: GOKULNATH

"""

#Step - 1  -Import Libraries and capture camera

#Step - 2  -Convert frames Into hsv

#Step - 3  -Track hand on color basis

#Step - 4  -Create mask on the basis of color and filter actual color

#Step - 5  -Invert pixel value and then enchance the result for better output

#Step - 6  -Find Contours for specific colored object

#Step - 7  -Find Max area contour and draw it on live feed

#Step - 8  -Find Convexity detect  for counting Values and Apply Cosin method

#Step - 9  -Put Counting Values on fingers on the basis of convexity defects.

#Step -10  -Enjoy Your Output



#Counting Fingers only tutorials using open cv


#Step 1---------------------------------------

import cv2 import

numpy as np

import math


#Read Camera
```

```
cap =
cv2.VideoCapture(0,cv2.CAP_DSHOW) def
nothing(x):     pass
#window name cv2.namedWindow("Color
Adjustments",cv2.WINDOW_NORMAL) cv2.resizeWindow("Color
Adjustments", (300, 300))  cv2.createTrackbar("Thresh", "Color
Adjustments", 0, 255, nothing)

#COlor Detection Track

cv2.createTrackbar("Lower_H", "Color Adjustments", 0, 255, nothing)
cv2.createTrackbar("Lower_S", "Color Adjustments", 0, 255, nothing)
cv2.createTrackbar("Lower_V", "Color Adjustments", 0, 255, nothing)
cv2.createTrackbar("Upper_H", "Color Adjustments", 255, 255, nothing)
cv2.createTrackbar("Upper_S", "Color Adjustments", 255, 255, nothing)
cv2.createTrackbar("Upper_V", "Color Adjustments", 255, 255, nothing)

while True:     _,frame = cap.read()     frame =
cv2.flip(frame,2)     frame =
cv2.resize(frame,(600,500))     # Get hand data from
the rectangle sub window     cv2.rectangle(frame,
(0,1), (300,500), (255, 0, 0), 0)     crop_image =
frame[1:500, 0:300]
```

```
    # Step  - 2 -
    hsv = cv2.cvtColor(crop_image, cv2.COLOR_BGR2HSV)
    #detecting hand     l_h = cv2.getTrackbarPos("Lower_H",
"Color Adjustments")     l_s =
cv2.getTrackbarPos("Lower_S", "Color Adjustments")     l_v
= cv2.getTrackbarPos("Lower_V", "Color Adjustments")

    u_h = cv2.getTrackbarPos("Upper_H", "Color Adjustments")
u_s = cv2.getTrackbarPos("Upper_S", "Color Adjustments")
u_v = cv2.getTrackbarPos("Upper_V", "Color Adjustments")

    #Step -3     lower_bound =
np.array([l_h, l_s, l_v])     upper_bound =
np.array([u_h, u_s, u_v])

    #step -4
    #Creating Mask     mask = cv2.inRange(hsv,
lower_bound, upper_bound)
    #filter mask with image     filtr =
cv2.bitwise_and(crop_image, crop_image, mask=mask)

    #step - 5     mask1  = cv2.bitwise_not(mask)    m_g = cv2.getTrackbarPos("Thresh",
"Color    Adjustments")    #getting    track    bar    value                    ret,thresh    =
cv2.threshold(mask1,m_g,255,cv2.THRESH_BINARY)                                dilata      =
cv2.dilate(thresh,(3,3),iterations = 6)
```

```
   #step - 6
   #findcontour(img,contour_retrival_mode,method)     cnts,hier =
cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)



try:
      #print("try")

      #Step -7
      # Find contour with maximum area        cm =
max(cnts, key=lambda x: cv2.contourArea(x))
      #print("C==",cnts)        epsilon =
0.0005*cv2.arcLength(cm,True)        data=
cv2.approxPolyDP(cm,epsilon,True)

      hull = cv2.convexHull(cm)

      cv2.drawContours(crop_image, [cm], -1, (50, 50, 150), 2)
cv2.drawContours(crop_image, [hull], -1, (0, 255, 0), 2)

      #Step -8
      # Find convexity defects
```

```
        hull = cv2.convexHull(cm, returnPoints=False)
    defects = cv2.convexityDefects(cm, hull)
    count_defects = 0
        #print("Area==",cv2.contourArea(hull) - cv2.contourArea(cm))
    for i in range(defects.shape[0]):
            s,e,f,d = defects[i,0]


            start = tuple(cm[s][0])
    end = tuple(cm[e][0])
    far = tuple(cm[f][0])


            #Cosin Rule            a = math.sqrt((end[0] - start[0]) ** 2 + (end[1] -
    start[1]) ** 2)            b = math.sqrt((far[0] - start[0]) ** 2 + (far[1] - start[1])
    ** 2)            c = math.sqrt((end[0] - far[0]) ** 2 + (end[1] - far[1]) ** 2)
    angle = (math.acos((b ** 2 + c ** 2 - a ** 2) / (2 * b * c)) * 180) / 3.14
            #print(angle)
            # if angle > 50 draw a circle at the far point
    if angle <= 50:
                count_defects += 1
    cv2.circle(crop_image,far,5,[255,255,255],-1)


        print("count==",count_defects)


        #Step - 9
```

```
        # Print number of fingers
    if count_defects == 0:

            cv2.putText(frame, "1", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255),2)
    elif count_defects == 1:
            cv2.putText(frame, "2", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
    elif count_defects == 2:

            cv2.putText(frame, "3", (5, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
    elif count_defects == 3:

            cv2.putText(frame, "4", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
    elif count_defects == 4:

            cv2.putText(frame, "5", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
          else:
    pass


       except:
    pass


       cv2.imshow("Thresh", thresh)
    #cv2.imshow("mask==",mask)
    cv2.imshow("filter==",filtr)
    cv2.imshow("Result", frame)     key =
```

cv2.waitKey(25) &0xFF if key == 27: break

cap.release() cv2.destroyAllWindows()

GitHub Link :

https://github.com/IBM-EPBL/IBM-Project-36716-1660297346

Project Demo Link:

https://drive.google.com/file/d/1D-2vKainjitiVjfF-lpuG0WRV0btEbdM/view?usp=share_link