A Gesture-based Tool for Sterile Browsing of Radiology Images

Team ID : PNT2022TMID54091

An IBM PROJECT REPORT

Submitted by

Dinesh Kumar S - (111919106008)
Udit Kumar B - (111919106063)
Balaji N - (111919106061)
Syed Farook M - (111919106002)
Eswar M - (111919106303)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

Electronics Communication Engineering

S.A ENGINEERING COLLEGE, CHENNAI - 600 077

(AUTONOMOUS)

ANNA UNIVERSITY : CHENNAI 600 025.

NOVEMBER 2022

INDEX

CHAPTER 1

INTRODUCTION

1.1 Project Overview

In this project we use gestures to browse images obtained during radiology. Gestures refer to non verbal form of communication made using hands.

A major challenge involved in this process is to provide doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work. Keyboards and pointing devices, such as a mouse, are today's principal method of human—computer interaction. However, the use of computer keyboards and mice by doctors and nurses in intensive care units (ICUs) is a common method for spreading infections. Humans can recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development.

In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others. In this project Gesture based Desktop automation, First the model is trained pre trained on the images of different hand gestures, such as a showing numbers with fingers as 1,2,3,4. This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the gesture predicts is 0 - 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 grayscale.

1.2 Purpose

It is used to browse through the images obtained using radiology using hand gestures rather than using mouse,keyboard,etc thereby maintaining sterility.

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.

CHAPTER 2
LITERATURE SURVEY

2.1 Existing problem

A major challenge involved in gesture process is to provide doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work. Keyboards and pointing devices, such as a mouse, are today's principal method of human—computer interaction. However, the use of computer keyboards and mice by doctors and nurses in intensive care units (ICUs) is a common method for spreading infections. Humans can recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development.

In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others.

2.2 References

S.No	Article	Authors	Published Year	Abstract

1	Literature Survey on Hand Gesture Recognition System	Akshatha, Bhavani Patil, Harshitha, Sindhu shree	February 2020	For those who are deaf and dumb Sign language is an efficient alternative way for talking, where we can understand them by using the hand gestures. For humans hands are a part of human organs which is used to manipulate physical objects. For this very reason hands are used most frequently by human beings to communicate and interact with machines. In the recent generation, hand gesture recognition system is improving in such a way that the interaction between the human and machine is advancing by using the electronic gadgets such as mobile phones, computers etc. So, there will be advancement not only in representing the speaking skills, also writing skills too. The real-time continuous gesture recognition is based on posture, position, orientation, and motion or by using the embedded systems like microcontrollers or it can be color maker approach, glove-based approach, vision-based approach and depth-based approach. The technique used in this system is that the input to the system will be given from the hand. They detect the image of the hand and pre-process it. Later on, they are going to crop the image how much they require for the analysis. In later stages they are going to extract the feature of the hand and then they are going to classify it. At the last the gesture is converted into the speech. According them hand gesture recognition system provides Human Computer Interaction. The two major applications they have used is Sign Language Recognition and gesture-based control.

2	Systematic literature review of hand gestures used in human computer interaction interfaces	ADEEN H.S, ATIA A, AMIN A, VICTOR A, ESSAM A, GHARIB E HUSSIEN	August 2015	There are three sub-types of iconic gestures: those that describe a shape (Pictographs), those that represent a spatial relation (Spatiographic), and those that describe action of an object (Kinematographs) (Rimé and Schiaratura, 1991). Metaphoric gestures "are iconic gestures which represent abstract content" (Wagner et al., 2014, McNeill, 1992), e.g. a cutting gesture to indicate a decision has been made (Casasanto and Lozano, 2007). They "sketch in space the logical track followed by the speaker's thinking" (Rimé and Schiaratura, 1991). Modalizing symbolic gestures primarily complement speech, but can also complement other means of communication.
3	HAND GESTURE RECOGNITION : A LITERATURE REVIEW	Rafiqul Zaman Khan Noor Adnan Ibraheem	July 2012	Hand gesture recognition system received great attention in the recent few years because of itsmanifoldness applications and the ability to interact with machine efficiently through human computer interaction. In this paper a survey of recent hand gesture recognition systems is presented. Key issues of hand gesture recognition system are presented with challenges of gesture system. Review methods of recent postures and gestures recognition system presented as well.

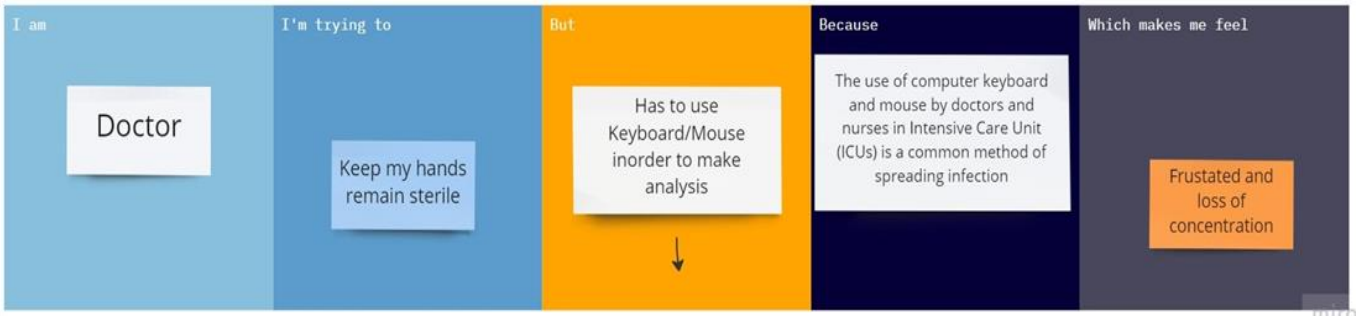
4	Real-Time Hand Gesture Interface for Browsing Medical Images	Juan Wachs, Helman Stern, Yae lEdan, Craig Feied, Mark Smith Jon	March 2007	A gesture interface is developed for users, such as doctors/surgeons, to browse medical images in a sterile medical environment. A vision-based gesture capture system 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 real-time 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.

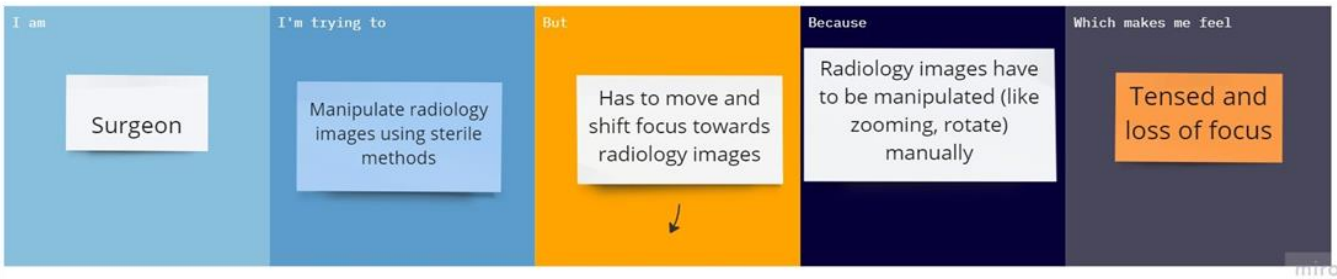
5	Gesture-controlled image system positioning for minimally invasive interventions	Hatsche rB, Mewes A, Pannicke E, Kagebein U, Wacker F, Hansen C, Hensel.	December 2020	This work examines how a touchless interaction concept contributes to an efficient, direct, and sterile interaction workflow during CT-guided interventions. Twohand gesture sets were designed specifically under consideration of the clinical workflow and the hardware capabilities. These were used to change the position of an X-Ray tube and detector of a CT scanner without breakingsterility and are compared regarding usability and performance in a user study with 10 users. The user study revealed that it ispossible to change the angle of the gantry within 10 secondsaverage in an experimental setup. A straight hand gesture showed higher acceptance than a pistol motivated gesture. Furthermore, the sequences were not optimal and confused the users. It turned out that it feels more natural to activate and confirm the system with the same gesture.

2.3 Problem Statement Definition

The Problem Statement helps you focus on what matters to create experiences people will love. A well-articulated problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers





Problem Statement (PS)	**Iam (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**

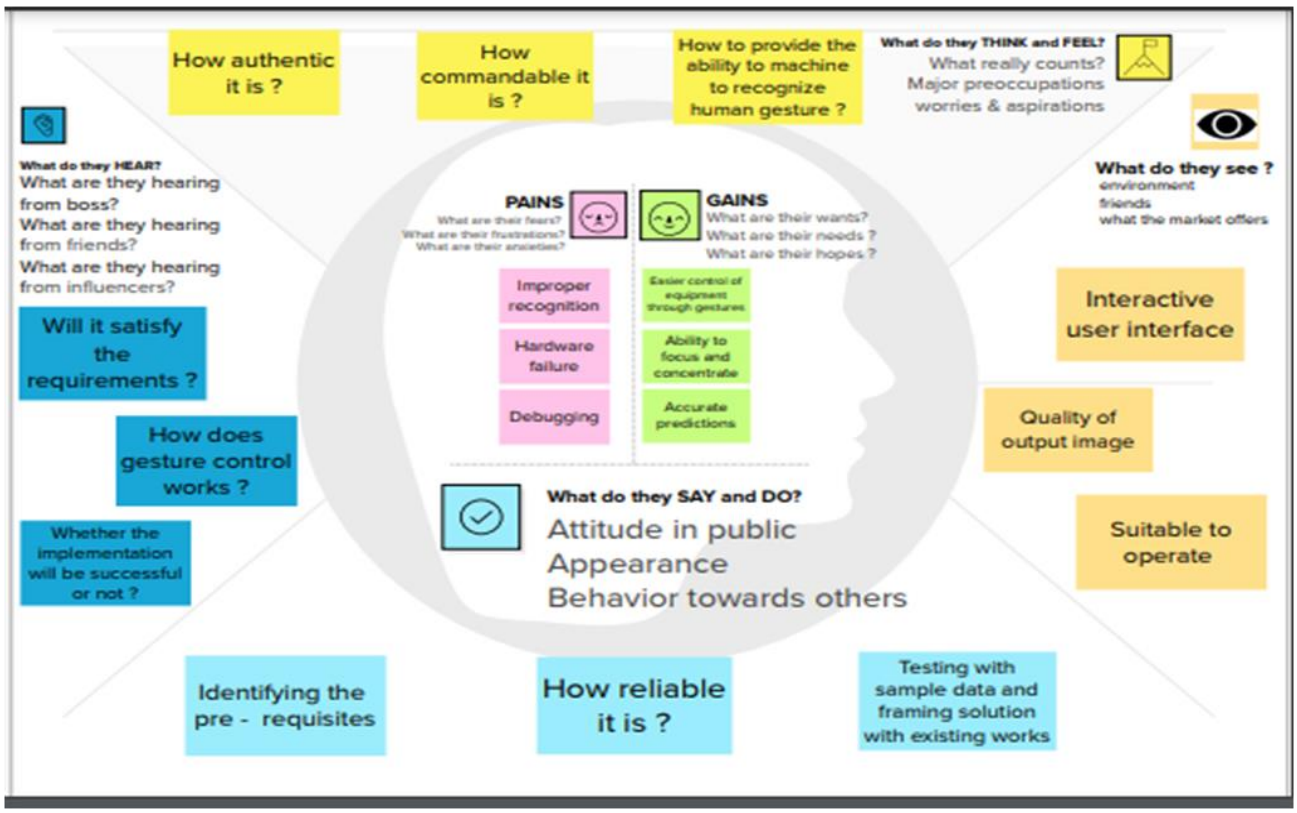
PS-1	Doctor	Keep my hands remain sterile	Has to use Keyboard/Mouse inorder to make analysis	The use of computer keyboard and mouse by doctors and nurses in Intensive Care Unit (ICUs) is a common method of spreading infection	Frustated and loss of concentration
PS-2	Surgeon	Manipulate radiology images using sterile methods during surgery	Has to move and shift focus towards radiology images	Radiology images have to be manipulated (like zooming, rotate) manually	Tensed and loss of focus
PS-3	Doctor/Surgeon	Manipulate radiology images using sterile methods	It disturbs the patients	Of the noise produced during noise recognition system	Annonying and frustrating

CHAPTER 3
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation & Brainstorming

Ideation

In order to provide surgeons with a more efficient, comfortable, precise, and sterile interaction technique, the hands can be an effective means of accomplishing this goal in comparison to other modalities, such as voice or eye interaction . Touch-less gesture interaction is an option to interact with imaging systems, displays, and controllers without breaking the sterility barrier. The

system utilizes nothing but a camera with good quality and can follow the hand of the user in 2 dimensions and identify up to four mouse-defined hand motions.

Recent progress in artificial intelligence provides innovative opportunities for motion tracking and human-machine interaction. In the field of healthcare, sensors like Microsoft Kinect has been used for detecting postures. And using electromyography technology to capture gesture instead of the camera, therefore it is less affected by the external factors such as light and obstruction.

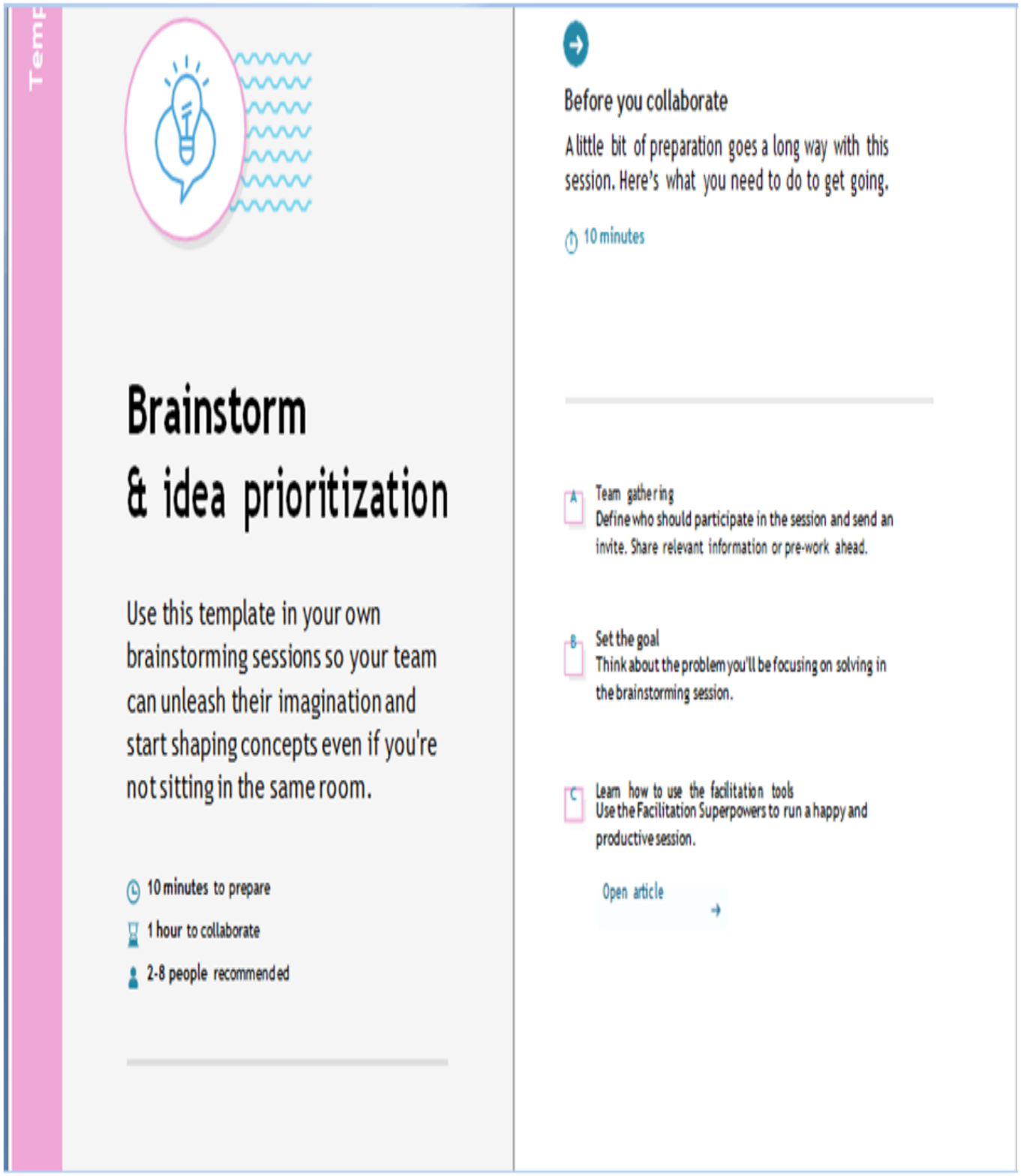
Voice command is another type of touchless communication but its commands are discrete rather than hand gestures which are able to perform analog commands. On the other hand, voice command has other disadvantages such as its low accuracy due to existence of noise in surgery rooms and accents.

Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

This helps the team to unleash their imagination and start shaping concepts even if you're not sitting in the same room.

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



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

The doctors' hands should be sterile inside the Operation Theatre. They would often need to pick some objects or would try to ask for some objects. They should be able to communicate without touching any objects which may lead to some infections. We try to avoid direct contact by recognizing the hand gestures of the doctors using an UI that is associated with an AI model.

Key rules of brainstorming

To run an smooth and productive session

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

Step-2: Brainstorm, Idea Listing and Grouping

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 and break it up into smaller sub-groups.

20 minutes

Gloves and Infections:

- It avoids infections
- Contactless
- Gloves should not affect the gestures meaning
- Less possibilities of new infections

Communication:

- The future gestures should not impact model
- The model should work at any kind of locations

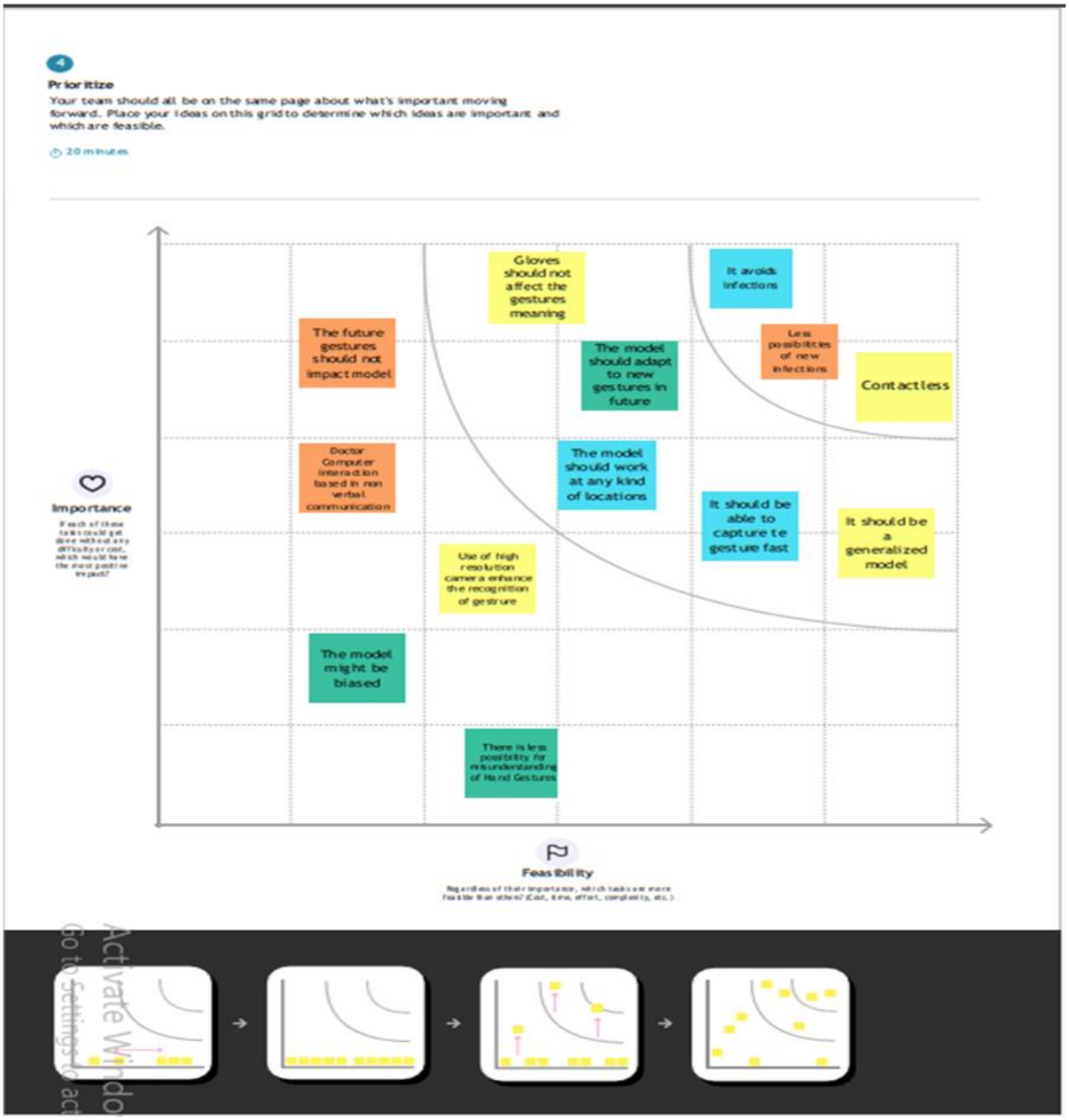
Model Complexities:

- The model might be biased
- The model should adapt to new gestures in future
- It should be a generalized model

Sensors and cameras:

- Doctor Computer interaction based in non verbal communication
- There is less possibility for misunderstanding of Hand Gestures
- It should be able to capture te gesture fast
- Use of high resolution camera enhance the recognition of gestrure

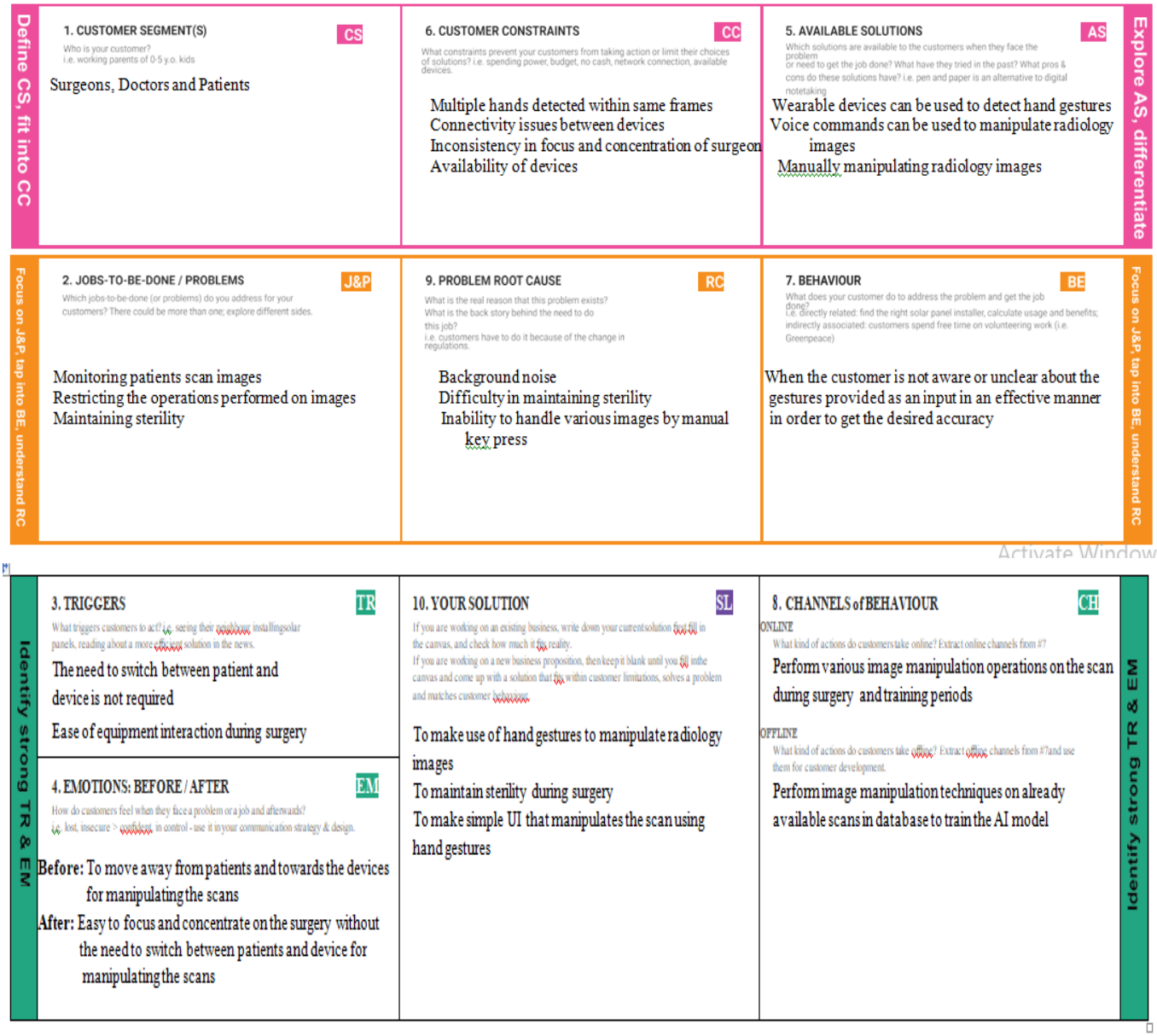
Step-3: Idea Prioritization



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Hand Gesture tool to do sterile navigation of radiology images
2.	Idea / Solution description	Use artificial intelligence technology to assist doctors by taking hand gestures as input and perform necessary actions on radiology images
3.	Novelty / Uniqueness	These Gestures helps to manipulate the radiology images and helps to stay focused for surgeons.
4.	Social Impact / Customer Satisfaction	The proposed system should provide a good manipulation of radiology images for surgeon during surgery supporting their focus of attention, and providing fast response times.
5.	Business Model (Revenue Model)	A Hand-based Gesture Recognition System used for detecting any kind of Gestures which when the given input Gesture matches with the trained image.
6.	Scalability of the Solution	The proposed approach allows the learning of new gestures with no need of recording real subjects.

3.4 Problem Solution fit



CHAPTER 4
REQUIREMENT ANALYSIS

4.1 Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Hand detection	Filtering of hand from video capturing device
FR-2	Filtered object detection	Reads and filters by recognizing clusters of skin coloured objects
FR-3	Gesture control	Hand gestures recognition for commands
FR-4	Hand calibration	Perform according to the adjustment of user's dominant hand
FR-5	Model rendering	When the user uploads/gives the gestures, the algorithm should start processing its task.
FR-6	Launching the model	Launch the application either from cloud where it is deployed or by installation but with a stable internet connectivity

4.2 Non-Functional requirements

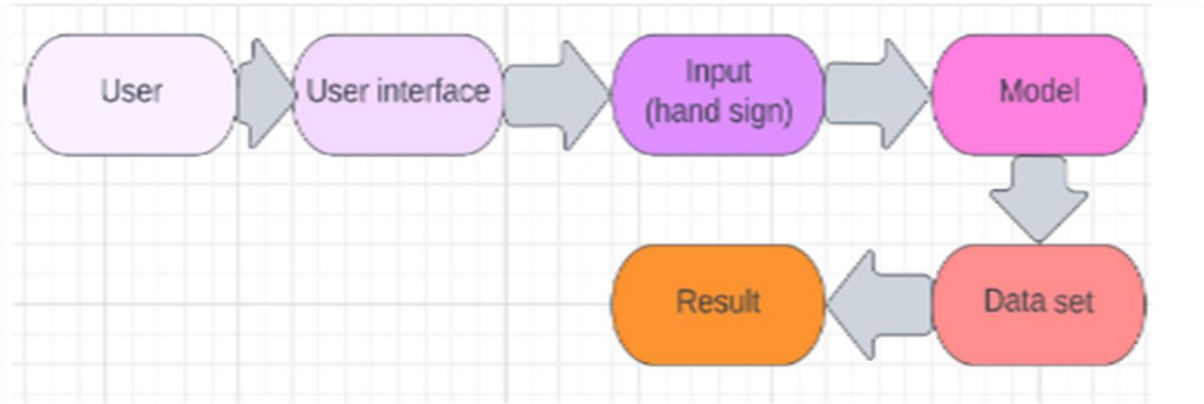
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Usability is easy for all users. It is understandable for non technical users with minimal instructions
NFR-2	Security	Accessible only in secure networks with administrative permissions, so there is less chance of security breach
NFR-3	Reliability	It is operable under all conditions, regardless of user's operating environment
NFR-4	Performance	Minimize the number of calculation to perform hand gesture and to improve image resolution quality
NFR-5	Availability	When the gesture is available then only the application works. This application is only available in surgery rooms
NFR-6	Scalability	Model is scaled by CNN with help of data augmentation and gesture recognition using OpenCV, Tensorflow, Keras

CHAPTER 5
PROJECT DESIGN

5.1 Data Flow Diagrams

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

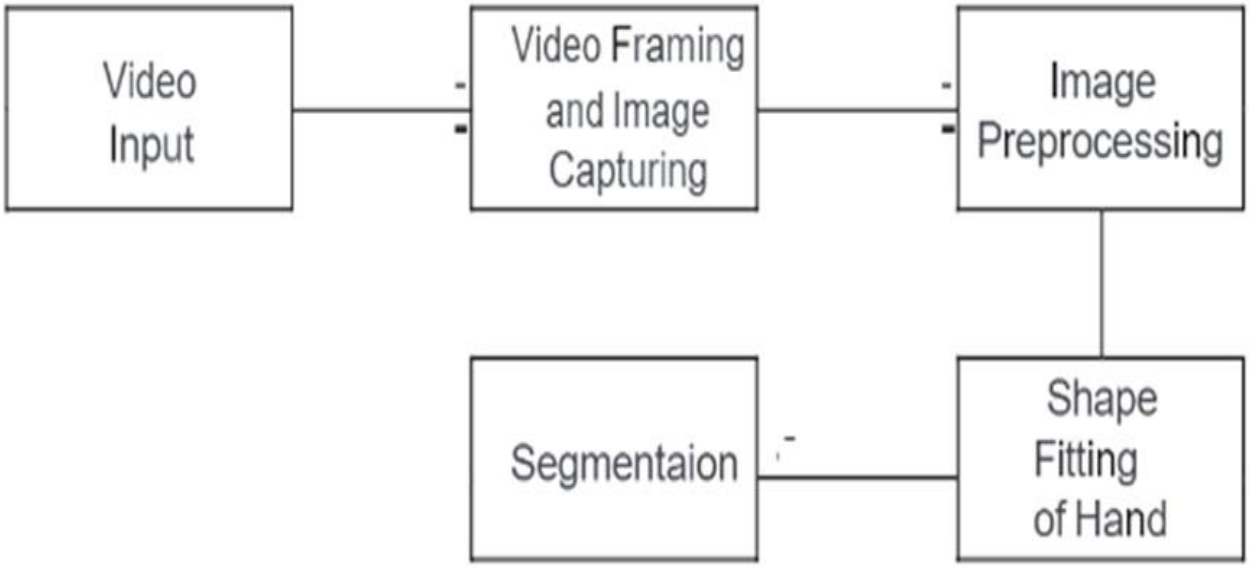
Simplified



Data Flow Diagram - Level 0

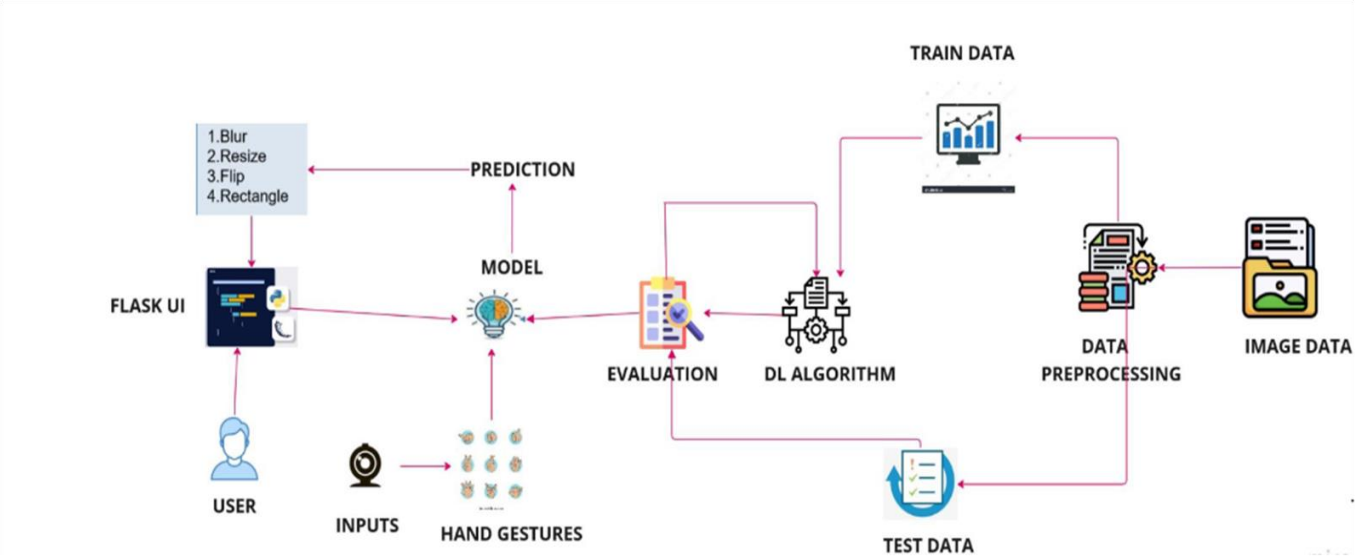
Data Flow Diagram - Level 1

Data Flow Diagram - Level 2



5.2 Solution & Technical Architecture

Technical Architecture



Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution architecture for a gesture based tool for sterile browsing of radiology images.

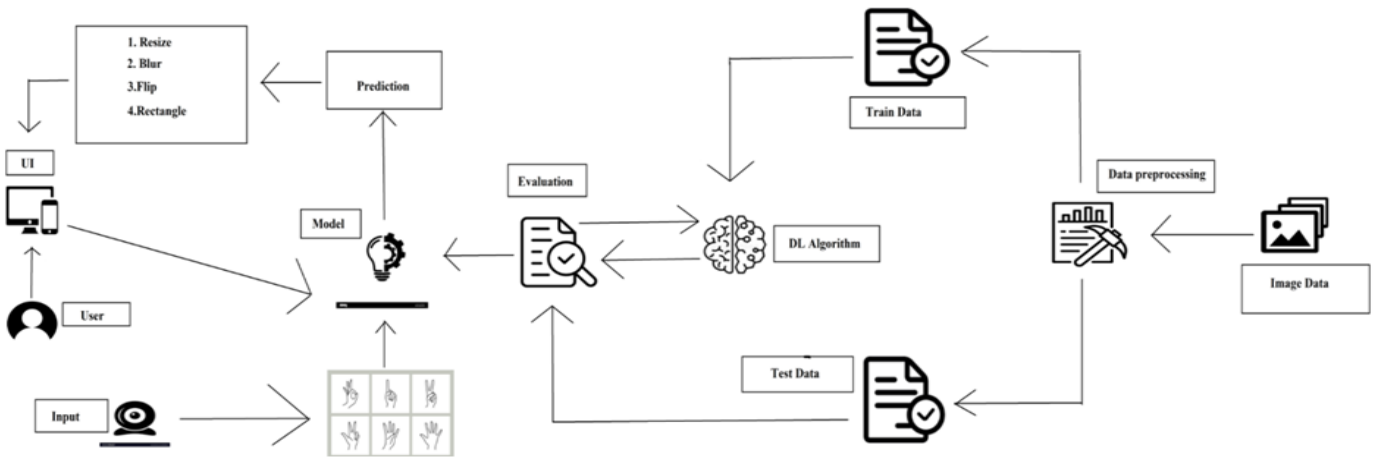
1. User (Doctor/Surgeon) is giving hand gestures as input to perform the certain actions such as zoom in, zoom out the image of the patients during the surgery.

2. In this project Gesture based project, First the model is trained, pre trained on the images of different hand gestures,

3. 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.

4. If the gesture predicate is 1 then images is resized into 200x200; 2. image is rotated to 45 degree right side; 3. image is blurred; 4. Image is resized into 400x400; 5. Image is converted to greyscale.

Solution Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer [Doctor/Surgeon] (Web user)	Launch	USN-1	As a user I can launch the webpage to upload and manipulate the scan images	I can access the webpage	High	Sprint-4

		USN-2	As a user I can use different web browsers	I can access the webpage using different web browsers	High	Sprint-1
Administrator	IBM Cloud	USN-1	Access the database	Database Management	High	Sprint-3
		USN-2	Server crash, database recovery	Resolve the errors/ issue, recover the lost data from database	High	Sprint-5
Customer care executive	Availability	USN-1	Interpret and recognize gesture inaccurately	Webcam detection	Medium	Sprint-5
		USN-2	When the website is unresponsive or an internal error occurs in the website	Webpage is unresponsive	Medium	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	Application/Software Launch	USN-1	As a user, I can launch the developed application/software	10	Medium	Dinesh Kumar S Syed Farook M Balaji N Udit Kumar B Eswar M
Sprint-1	Accessing the User Interface (UI)	USN-2	As a user, I can interact with software and operate the application with the help of UI	10	Medium	Dinesh Kumar S Syed Farook M Balaji N Udit Kumar B Eswar M
Sprint-2	Launching the webcam/camera	USN-3	As a user, I can open the webcam/camera from the application to perform gestures	12	Low	Dinesh Kumar S Syed Farook M Balaji N Udit Kumar B Eswar M

Sprint-2	Upload images from local system for manipulation	USN-4	As a user, I can upload images to the application from local system for manipulation	18	Low	Dinesh Kumar S Syed Farook M Balaji N Udit Kumar B Eswar M
Sprint-3	Manipulating images through gestures	USN-5	As a user, I can perform various gestures with respect to system specification to manipulate the images	20	Medium	Dinesh Kumar S Syed Farook M Balaji N Udit Kumar B Eswar M
Sprint-4	Display the result/output	USN-6	As a user, I can see the sterile browsed/manipulated image on the screen with respect to the gesture performed	20	High	Dinesh Kumar S Syed Farook M Balaji N Udit Kumar B Eswar M

6.2 Sprint Delivery Schedule

SI. NO	**MILESTONE**	**ACTIVITIES**	**DATE**
1	Preparation Phase	Pre-requisites	24 Aug 2022
		Prior knowledge	25 Aug 2022
		Project Structure	23 Aug 2022

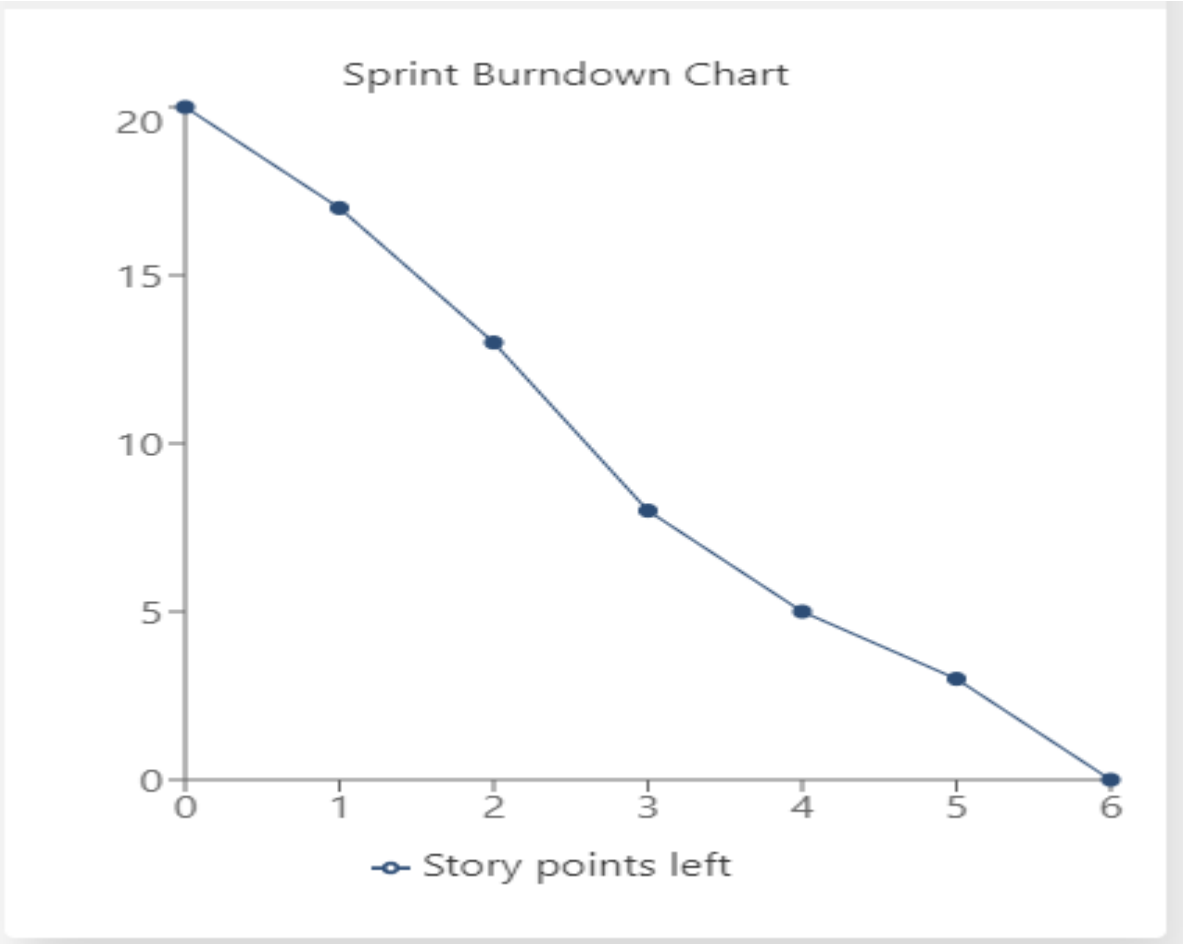
		Project Flow	23 Aug 2022
		Project Objectives	22 Aug 2022
		Registrations	26 Aug 2022
		Environment Set-up	27 Aug 2022
2	Ideation Phase	Literature Survey	29 Aug 2022 - 03 Sept 2022
		Empathy Map	5 Sept 2022 - 7 Sept 2022
		Problem Statement	8 Sept 2022 – 10 Sept 2022
		Ideation	12 Sept 2022- 16 Sept 2022
3	Project Design Phase -I	Proposed Solution	19 Sept 2022 – 23 Sept 2022
		Problem Solution Fit	24 Sept 2022 – 26 Sept 2022
		Solution Architecture	27 Sept 2022 – 30 Sept 2022
4	Project Design Phase -II	Customer Journey	3 Oct 2022 – 8 Oct 2022
		Requirement Analysis	9 Oct 2022 – 11 Oct 2022
		Data Flow Diagrams	11 Oct 2022 – 14 Oct 2022
		Technology Architecture	15 Oct 2022 – 16 Oct 2022
5	Project Planning Phase	Milestones & Tasks	17 Oct 2022 – 18 Oct 2022
		Sprint Schedules	19 Oct 2022- 22 Oct 2022

6	Project Development Phase	Sprint-1	24 Oct 2022 - 29 Oct 2022
		Sprint-2	31 Oct 2022 - 05 Nov 2022
		Sprint-3	07 Nov 2022 - 12 Nov 2022
		Sprint-4	14 Nov 2022 - 19 Nov 2022

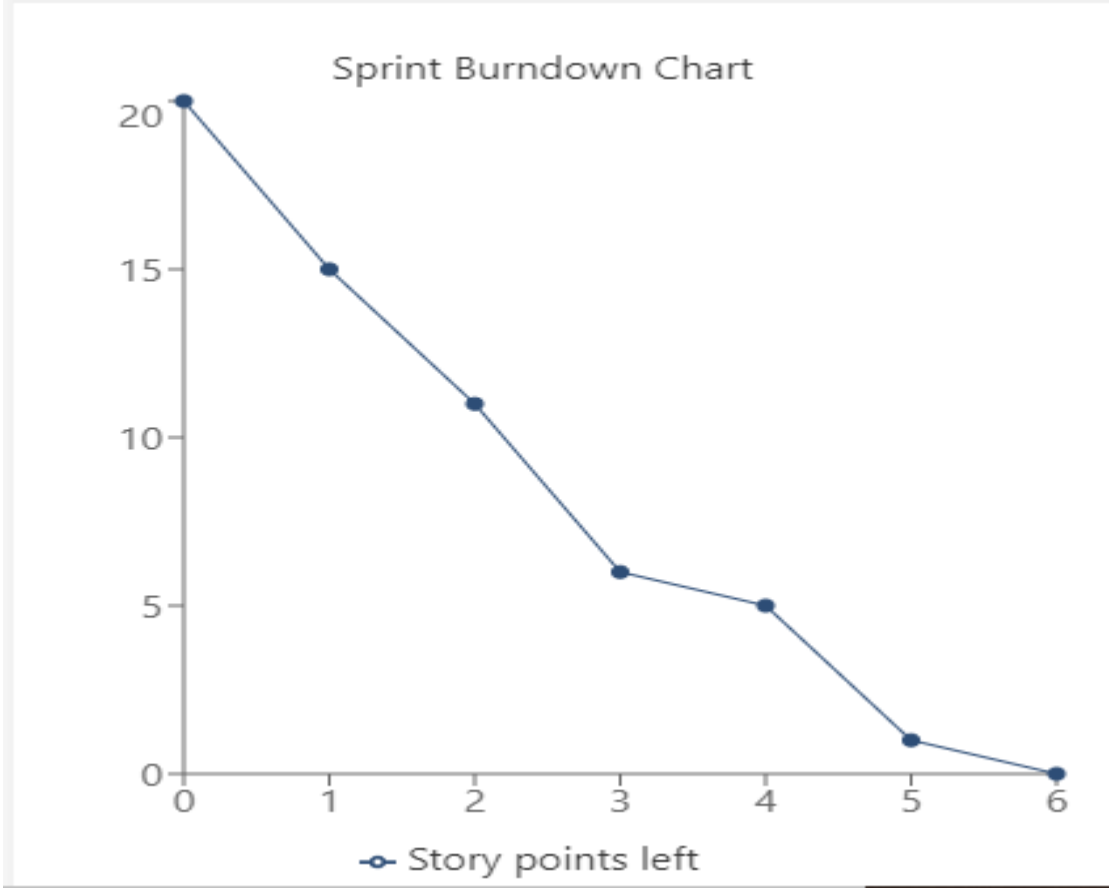
6.3 Reports from JIRA

Burndown Chart

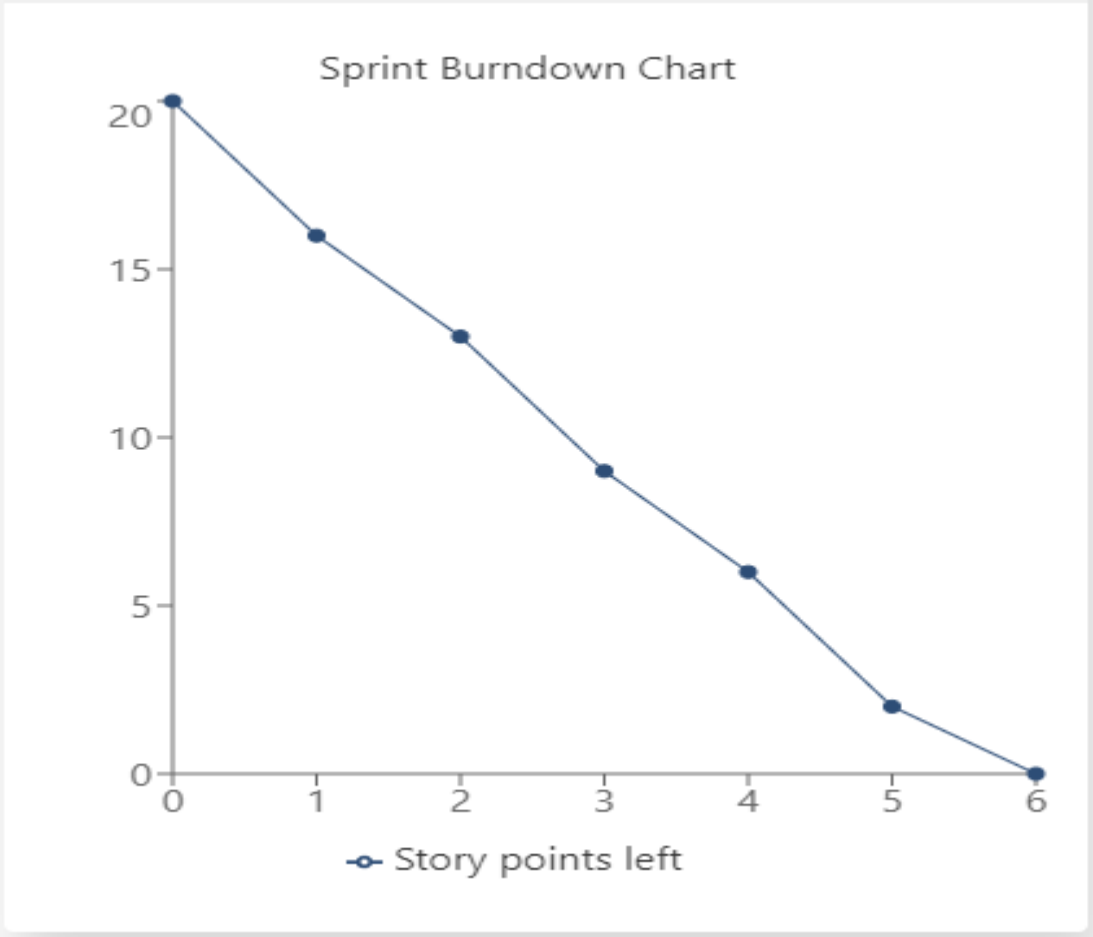
Burndown Chart for Sprint 1



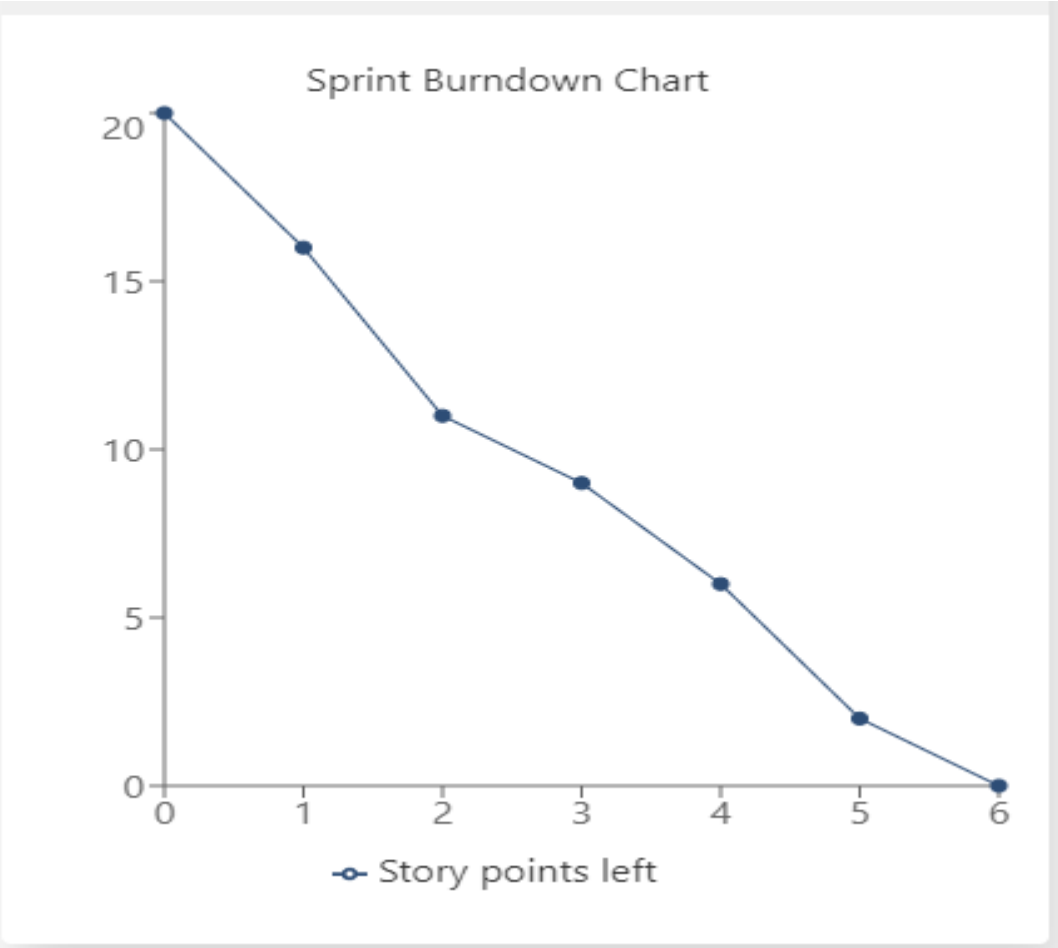
Burndown Chart for Sprint 2



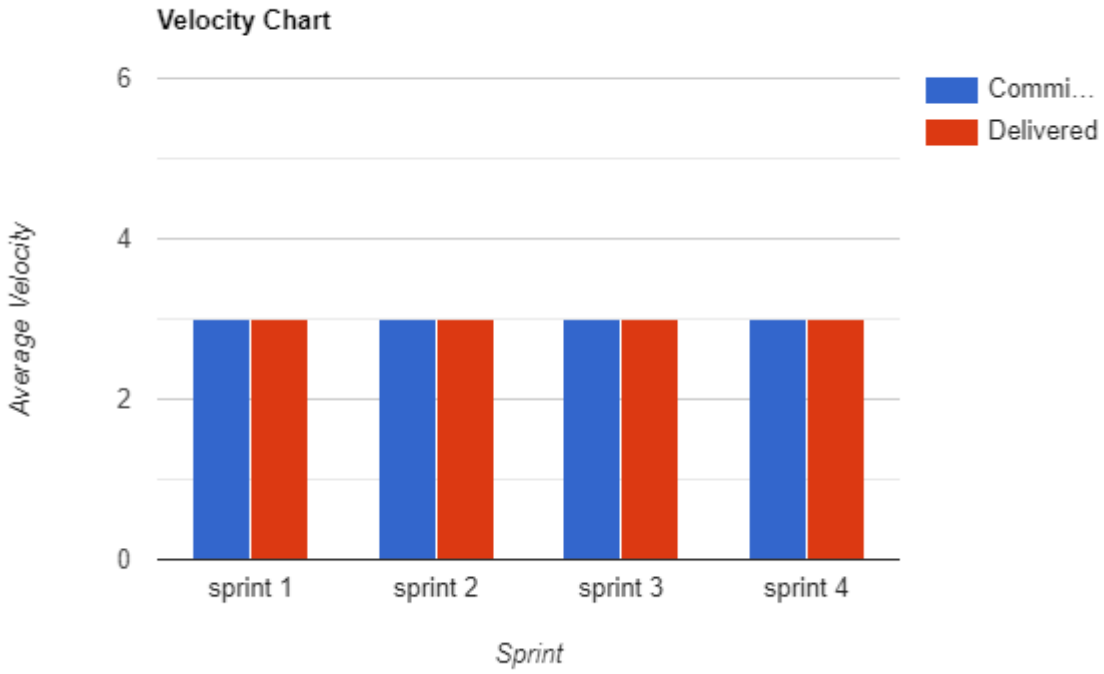
Burndown Chart Sprint 3



Burndown Chart Sprint 4

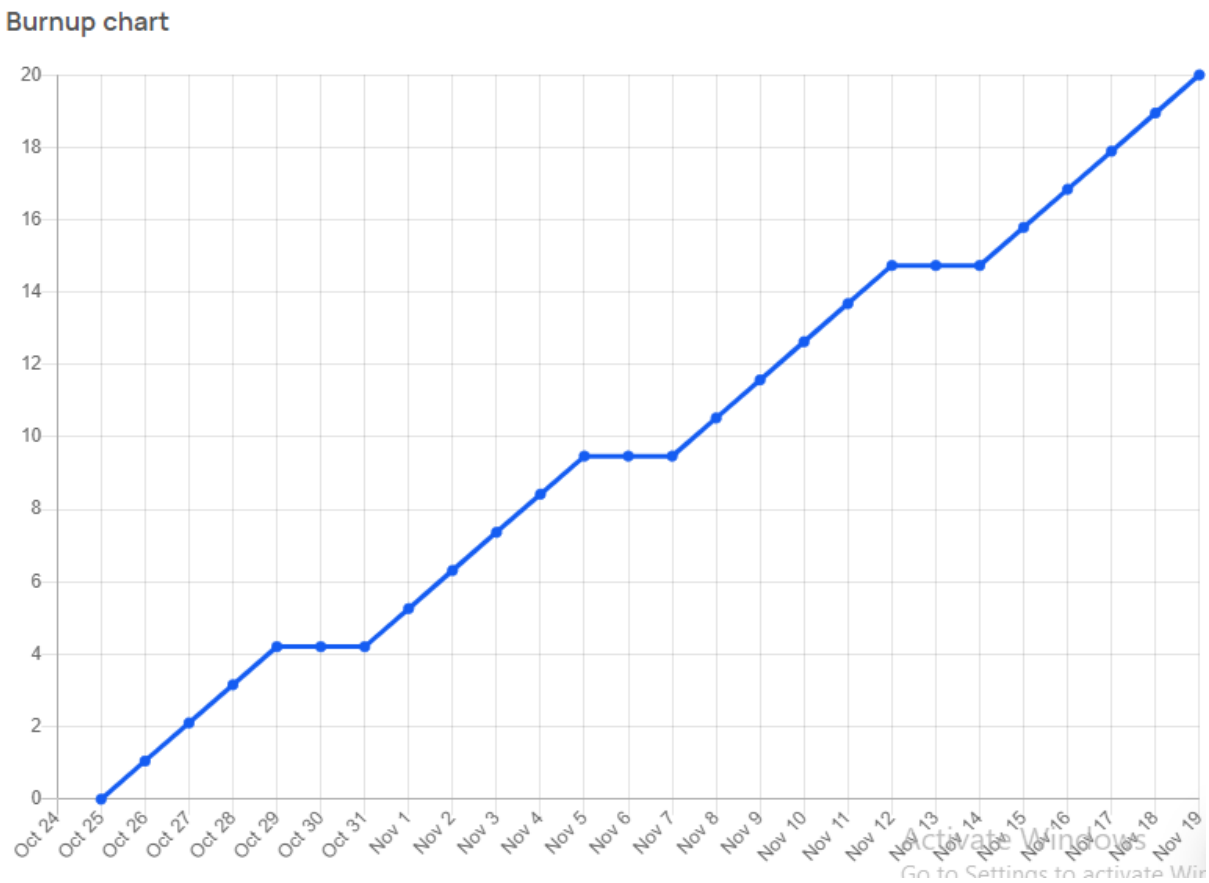


Velocity Chart



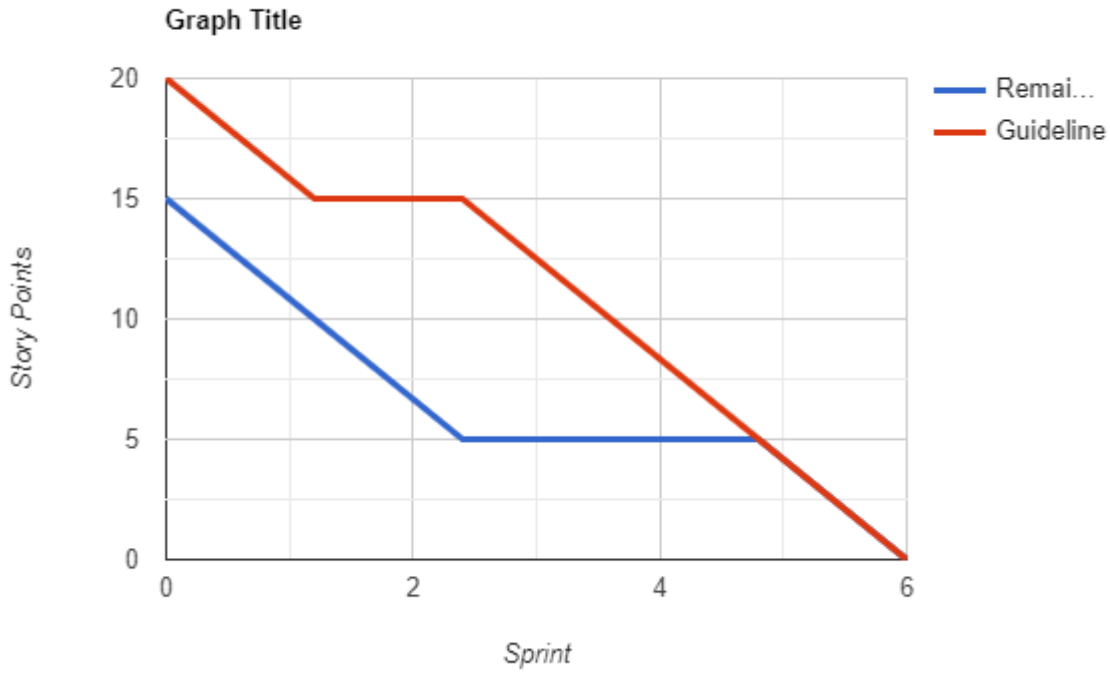
Burnup Chart

Burnup Chart for Sprint 1 - Sprint 4

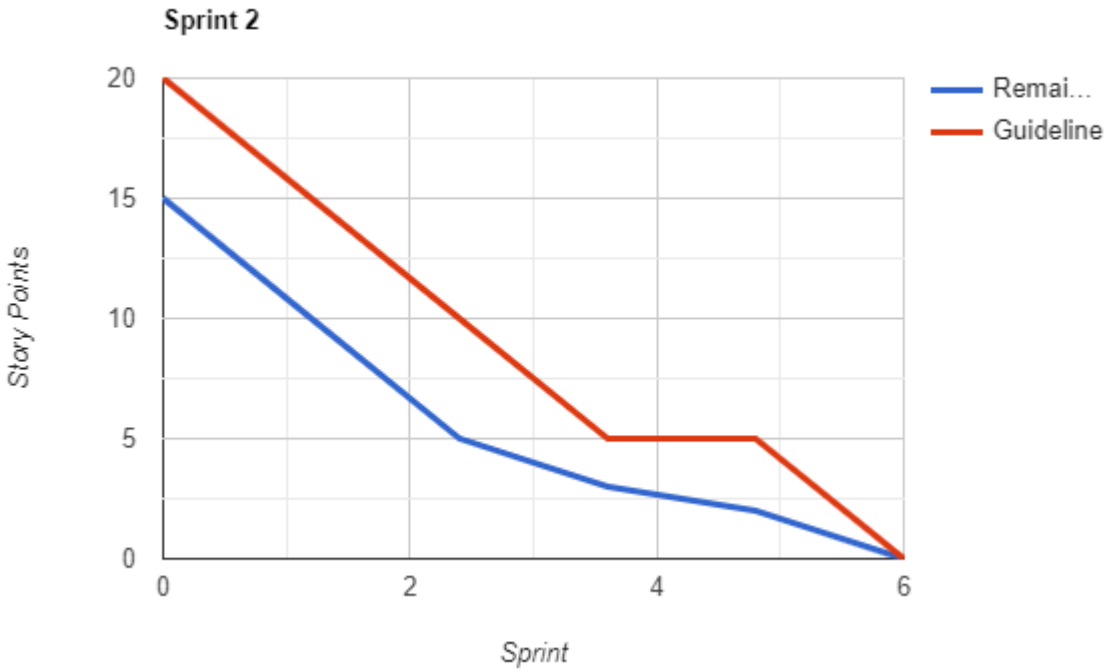


Sprint Report

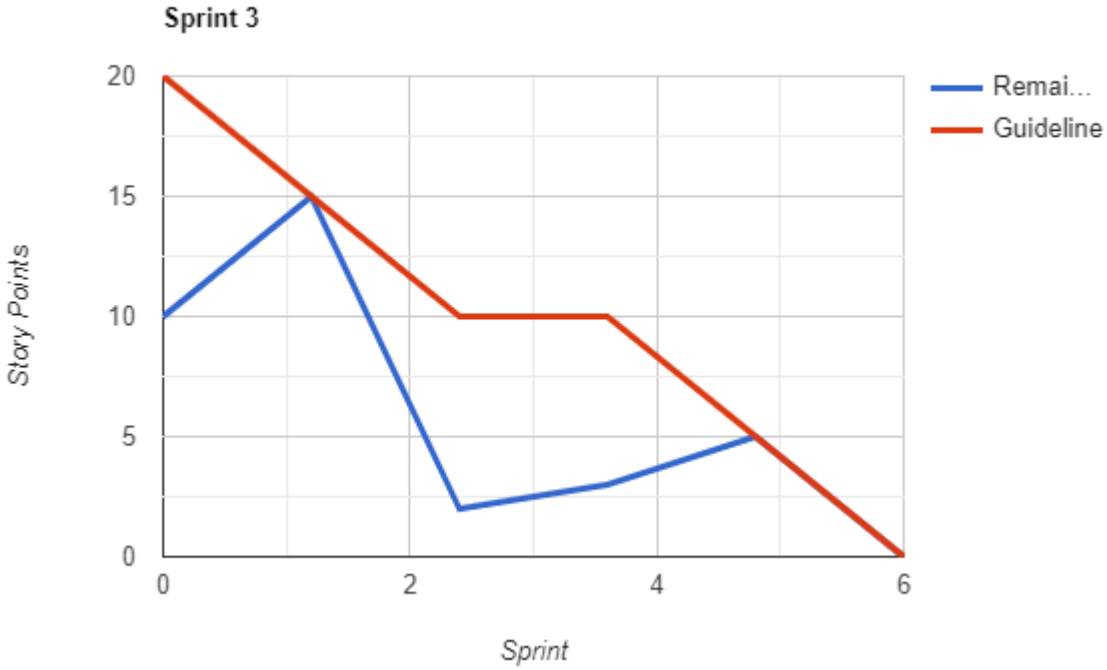
Sprint 1



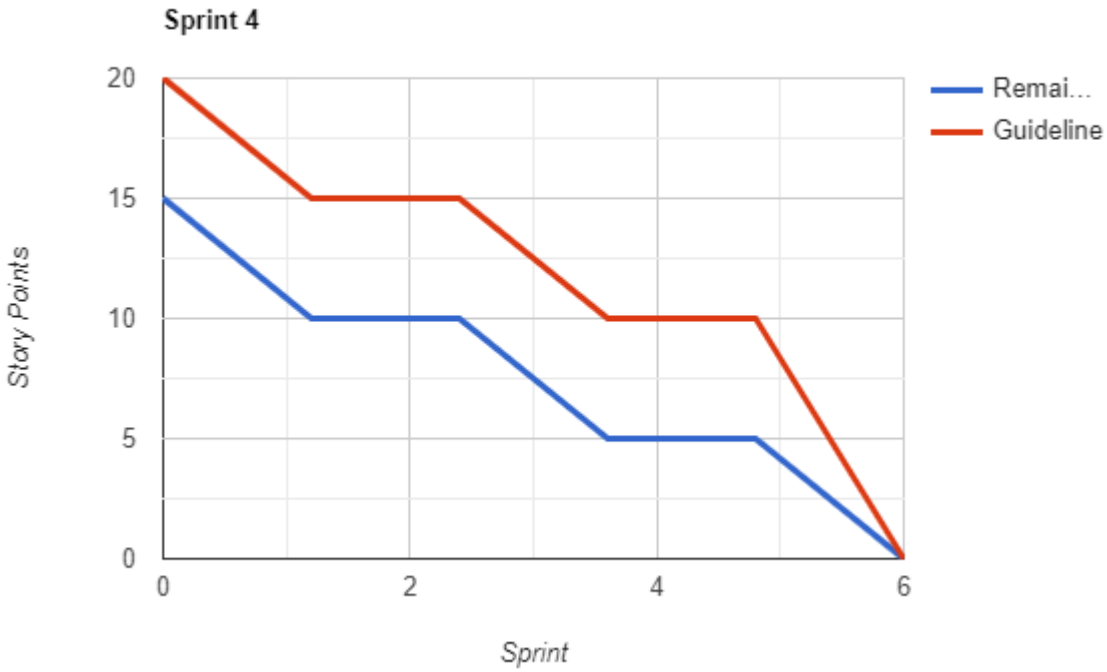
Sprint 2



Sprint 3



Sprint 4


Sprint 4
20
15
10
5
0
0
2
4
6
Story Points
Sprint
Remai...
Guideline

CHAPTER 7

CODING & SOLUTIONING

7.1 Demo video

The demo video is found on the demo page and can be accessed through the link https://127.0.0.1:5000/demo or https://127.0.0.1:5000/ and click on demo

This demo video showcases the browsing of radiology images through sterile methodology of gesture based

```
<iframe width="780" height="440"
src="https://www.youtube.com/embed/nD621G8u6oc?start=3&loop=1&autoplay=1&mute=1&controls=0">
</iframe><br/>
```

7.2 Home header

The home header is found on the top of home page and can be accessed through the link https://127.0.0.1:5000/

This home header leads to the smartinternzl page of the projet when the header is clicked

```
<a href="https://smartinternz.com/guided-project/a-gesture-based-tool-for-sterile-browsing-of-radiology-images-cnn-and-open-cv">
            <h2> Gesture-based Tool for Sterile Browsing of Radiology Images</h2>

        </a>
```

7.3 Upload the image

The upload image feature is found in launch page and can be accessed through the link https://127.0.0.1:5000/launch or https://127.0.0.1:5000/ and click on launch

This feature is used to upload the image which is going to be manipulated

```
<form action = "http://localhost:5000/" id="upload-file" method="post" enctype="multipart/form-data">
```

```
        <br>

        <div class="upload">

        <label><i class="fa fa-upload" style="font-size: 50px; "aria-hidden="true"><input type="file" style="display:none;" name="image" id="imageUpload" accept=".png, .jpg, .jpeg,.pdf"></i></label>

        </div><br><br>

      </form>
```

```
.upload{

   border: 1px solid black;

   border-radius: 50%;

   padding: 20px;

   background-color: white;

 }
```

```
$(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);
    });
```

7.4 Manipulate button

It is found in launch page and can be accessed via https://127.0.0.1:5000/ and click on launch or https://127.0.0.1:5000/launch

It is used to start the manipulation of uploaded image through opencv2

```
<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">Manipulate</button>
        </div>
      </div>
      <div class="loader" style="display:none;"></div>
    </div>
    </div1>
        <footer>
    <script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
  </footer>
```

```
.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;
}
.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;
```

```
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;
}
```

```
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
```

```
  // 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: '/perform',
            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!');
            },
        });
    });
});
```

7.5 Resize image to 200x200 size

This feature resizes the image size to 200x200. This is achieved by showing 1 finger infront of webcam while manipulating the image

```
if prediction[0][0]=='ONE':
        resized = cv2.resize(image1, (200, 200))
        cv2.imshow("Fixed Resizing", resized)
        key=cv2.waitKey(3000)
        if (key & 0xFF) == ord("1"):
            cv2.destroyWindow("Fixed Resizing")
```

7.6 Rotate image by 45 degree right

This feature rotates the image 45 degree to right. This is achieved by showing 2 fingers infront of webcam while manipulating the image

```
elif prediction[0][0]=='TWO':
        (h, w, d) = image1.shape
        center = (w // 2, h // 2)
        M = cv2.getRotationMatrix2D(center, -45, 1.0)
        rotated = cv2.warpAffine(image1, M, (w, h))
        cv2.imshow("OpenCV Rotation", rotated)
        key=cv2.waitKey(3000)
        if (key & 0xFF) == ord("2"):
            cv2.destroyWindow("OpenCV Rotation")
```

7.7 Blur the image

This feature blurs the uploaded image. This is achieved by showing 3 fingers infront of webcam while manipulating the image

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

```
            blurred = cv2.GaussianBlur(image1, (21, 21), 0)

            cv2.imshow("Blurred", blurred)

            key=cv2.waitKey(3000)

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

                cv2.destroyWindow("Blurred")
```

7.8 Resize the image to 400x400 size

This feature resizes the uploaded image to the size 0f 400x400. This is achieved by showing 4 fingers infront of webcam while manipulating the image

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

```
            resized = cv2.resize(image1, (400, 400))

            cv2.imshow("Fixed Resizing", resized)

            key=cv2.waitKey(3000)

            if (key & 0xFF) == ord("4"):

                cv2.destroyWindow("Fixed Resizing")
```

7.9 Convert the image to grayscale

This feature converts the uploaded image to grayscale image. This is achieved by showing 5 fingers infront of webcam while manipulating the image

```
elif prediction[0][0]=='FIVE':
```

```
        gray = cv2.cvtColor(image1, cv2.COLOR_RGB2GRAY)

        cv2.imshow("OpenCV Gray Scale", gray)

        key=cv2.waitKey(3000)

        if (key & 0xFF) == ord("5"):

            cv2.destroyWindow("OpenCV Gray Scale")
```

CHAPTER 8

TESTING

8.1 Test Cases

Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N	BUG ID	Execut
UI	Home Page	Verify user is able to see and interact with home page		1.Enter URL 2.Check whether the home page is rendered	https://127.0.0.1:5000/	Home page should display	Working as expected	Pass		No		Justus
Functional	Home Page	Verify the UI header element in home page		1.Enter URL 2.Check whether the home page is rendered 3.Click on the header to see if it lands to other page	https://127.0.0.1:5000/	Details of smartinternz project page appears	Working as expected	Pass		No		Rama Justus
UI	Demo page	Verify user is able to see and interact with demo page		1.Enter URL and click on demo 2.Check whether the demo page is rendered	https://127.0.0.1:5000/demo https://127.0.0.1:5000/	Demo page should display	Working as	Pass		No		Prave
Functional	Demo page	Verify user is able to watch the demo video and manipulate as a normal video player		1.Enter URL(https://127.0.0.1:5000/) and click demo 2.Check whether the demo page is rendered 3.Click on the demo video to play it 4.Check whether the video plays 5.Check whether the webpage behaves like normal video player	https://127.0.0.1:5000/demo https://127.0.0.1:5000/	Demo video should run and the video should be manipulated as a normal video player	Working as expected	Pass		No		Prave Prave
UI	Launch page	Verify user is able to see and interact with launch page		1.Enter URL(https://127.0.0.1:5000/) and click launch 2.Check whether the launch page is rendered	https://127.0.0.1:5000/ https://127.0.0.1:5000/launch	Launch page should display	Working as	Pass		No		Prave
Functional	Launch page	Verify user is able to upload the image to be manipulated		1.Enter URL(https://127.0.0.1:5000/) and click launch 2.Check whether the launch page is rendered 3.Click on the file upload icon and upload the image 4.Click on the manipulate button to start manipulating the image	https://127.0.0.1:5000/ https://127.0.0.1:5000/launch Image to be manipulated	Application should show upload the image to be manipulated and manipulation operation should begin	Working as expected	Pass		No		Prave Justus
Functional	Launch page	Verify user is able to resize the image to 400x400 size	Image needs to uploaded which is going to be manipulated	1.Enter URL(https://127.0.0.1:5000/) and click launch 2.Check whether the launch page is rendered 3.Click on the file upload icon and upload the image 4.Click on the manipulate button to start manipulating the image 5.Show 4 (four fingers) infront of camera as input 6.Check whether the image is resized into 400x400	https://127.0.0.1:5000/ https://127.0.0.1:5000/launch Image to be manipulated	Application should show resized image of 400x400	Working as expected	Pass		Yes		Justus Prave
Functional	Launch page	Verify user is able to blur the image	Image needs to uploaded which is going to be manipulated	1.Enter URL(https://127.0.0.1:5000/) and click launch 2.Check whether the launch page is rendered 3.Click on the file upload icon and upload the image 4.Click on the manipulate button to start manipulating the image 5.Show 3 (three fingers) infront of camera as input 6.Check whether the image is blurred	https://127.0.0.1:5000/ https://127.0.0.1:5000/launch Image to be manipulated	Application should show blurred image	Working as expected	Pass		Yes		Prave Rama
Functional	Launch page	Verify user is able to resize the image to 200x200 size	Image needs to uploaded which is going to be manipulated	1.Enter URL(https://127.0.0.1:5000/) and click launch 2.Check whether the launch page is rendered 3.Click on the file upload icon and upload the image 4.Click on the manipulate button to start manipulating the image 5.Show 1 (one finger) infront of camera as input 6.Check whether the image is resized into 200x200	https://127.0.0.1:5000/ https://127.0.0.1:5000/launch Image to be manipulated	Application should show resized image of 200x200	Working as expected	Pass		Yes		Rama Prave
Functional	Launch page	Verify user is able to rotate the image 45 degree right	Image needs to uploaded which is going to be manipulated	1.Enter URL(https://127.0.0.1:5000/) and click launch 2.Check whether the launch page is rendered 3.Click on the file upload icon and upload the image 4.Click on the manipulate button to start manipulating the image 5.Show 2 (two fingers) infront of camera as input 6.Check whether the image is turned 45 degree to right	https://127.0.0.1:5000/ https://127.0.0.1:5000/launch Image to be manipulated	Application should show image turned 45 degree right	Working as expected	Pass		Yes		Justus
Functional	Launch page	Verify user is able to convert the image to grayscale	Image needs to uploaded which is going to be manipulated	1.Enter URL(https://127.0.0.1:5000/) and click launch 2.Check whether the launch page is rendered 3.Click on the file upload icon and upload the image 4.Click on the manipulate button to start manipulating the image 5.Show 5 (five fingers) infront of camera as input 6.Check whether the image is converted to grayscale	https://127.0.0.1:5000/ https://127.0.0.1:5000/launch Image to be manipulated	Application should show grayscaled image	Working as expected	Pass		Yes		Prave

S.no	Test Scenarios
1	Verify user is able to access the website
2	Verify user is able to navigate between home, introduction/demo and launch page
3	Verify user is able to watch the demo video in demo page
4	Verify user is able to upload the image to be manipulated
5	Veriify user is able to manipulate the uploaded image

S.no	Manipulation
1	Verify user is able to rotate image 45 degree right
2	Verify user is able to resize the image to 400x400
3	Verify user is able to convert the image into grayscale
4	Verify user is able to resize the image to 200x200
5	Verify user is able to blur the image

8.2 User Acceptance Testing

Purpose

It is to briefly explain the test coverage and open issues of the A Gesture-based Tool for Sterile Browsing of Radiology Image project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	6	4	3	5	18
Duplicate	1	2	1	0	4
External	2	3	0	2	7

Fixed	11	2	3	13	29
Not Reproduced	0	1	1	1	3
Skipped	0	0	2	2	4
Won't Fix	0	4	2	2	8
Totals	20	16	12	25	73

Test Case Analysis

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

Section	TotalCases	Not Tested	Fail	Pass
Print Engine	9	0	0	9
Client Application	45	0	5	40
Security	5	0	1	4
Outsource Shipping	3	0	0	3
Exception Reporting	8	2	1	5

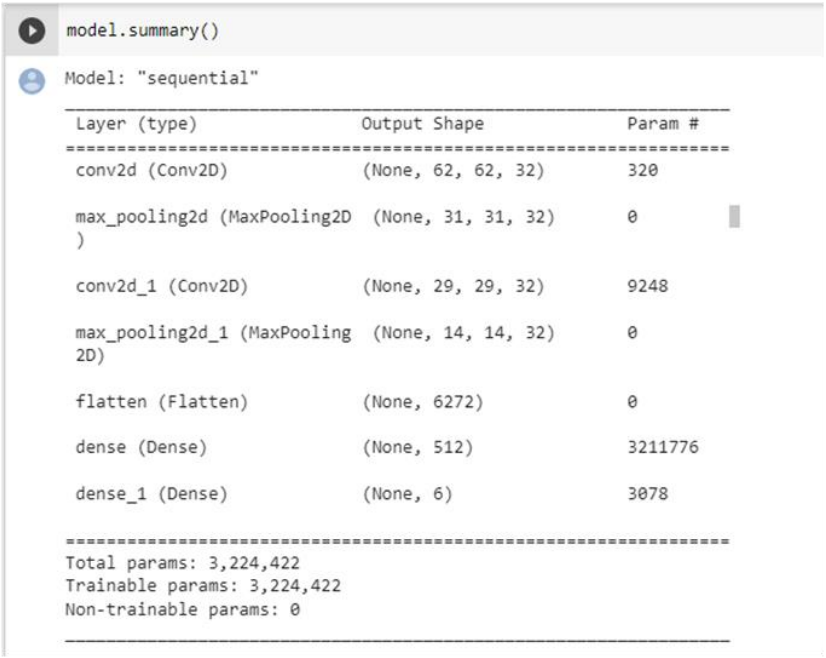
Final Report Output	10	1	2	7
Version Control	3	0	0	3

CHAPTER 9

RESULTS

9.1 Performance Metrics

Model Performance Testing

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	

```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        320

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 512)               3211776

 dense_1 (Dense)             (None, 6)                 3078

=================================================================
Total params: 3,224,422
Trainable params: 3,224,422
Non-trainable params: 0
_________________________________________________________________
```

2.	Accuracy	Training Accuracy - 0.9798 Validation Accuracy - 0.9667	Epoch 13/25 198/198 [==========] - 14s 70ms/step - loss: 0.0864 - accuracy: 0.9731 - val_loss: 0.2522 - val_accuracy: 0.9667 Epoch 14/25 198/198 [==========] - 15s 74ms/step - loss: 0.0518 - accuracy: 0.9781 - val_loss: 0.3543 - val_accuracy: 0.9333 Epoch 15/25 198/198 [==========] - 13s 66ms/step - loss: 0.0781 - accuracy: 0.9764 - val_loss: 0.3312 - val_accuracy: 0.9667 Epoch 16/25 198/198 [==========] - 13s 66ms/step - loss: 0.0487 - accuracy: 0.9832 - val_loss: 0.2124 - val_accuracy: 0.9667 Epoch 17/25 198/198 [==========] - 13s 66ms/step - loss: 0.0385 - accuracy: 0.9865 - val_loss: 0.2988 - val_accuracy: 0.9667 Epoch 18/25 198/198 [==========] - 13s 66ms/step - loss: 0.0573 - accuracy: 0.9832 - val_loss: 0.2565 - val_accuracy: 0.9667 Epoch 19/25 198/198 [==========] - 13s 66ms/step - loss: 0.0281 - accuracy: 0.9865 - val_loss: 0.2607 - val_accuracy: 0.9667 Epoch 20/25 198/198 [==========] - 13s 66ms/step - loss: 0.0501 - accuracy: 0.9848 - val_loss: 0.5327 - val_accuracy: 0.8667 Epoch 21/25 198/198 [==========] - 15s 74ms/step - loss: 0.0513 - accuracy: 0.9832 - val_loss: 0.3753 - val_accuracy: 0.9667 Epoch 22/25 198/198 [==========] - 13s 67ms/step - loss: 0.0369 - accuracy: 0.9882 - val_loss: 0.3396 - val_accuracy: 0.9667 Epoch 23/25 198/198 [==========] - 13s 68ms/step - loss: 0.0276 - accuracy: 0.9916 - val_loss: 0.4354 - val_accuracy: 0.9667 Epoch 24/25 198/198 [==========] - 15s 74ms/step - loss: 0.0752 - accuracy: 0.9697 - val_loss: 0.7543 - val_accuracy: 0.8667 Epoch 25/25 198/198 [==========] - 13s 67ms/step - loss: 0.0489 - accuracy: 0.9798 - val_loss: 0.2213 - val_accuracy: 0.9667 <keras.callbacks.History at 0x7f5cb5795dd0>

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.
- The Hand Gesture Recognition system provides a natural way of interfacing with the computers; hence it is more User friendly..
- There is less wear and tear of the computer as the standard input devices are eliminated and a camera is used as an input device.

Disadvantages

- The tool can be quite expensive as it requires cameras and other expensive devices to capture images and process it.
- Such systems are difficult to develop because of their complexity and their cost of implementation.
- As each gesture is assigned a specific control command, this system is not platform independent since certain control commands vary as the operating system varies.

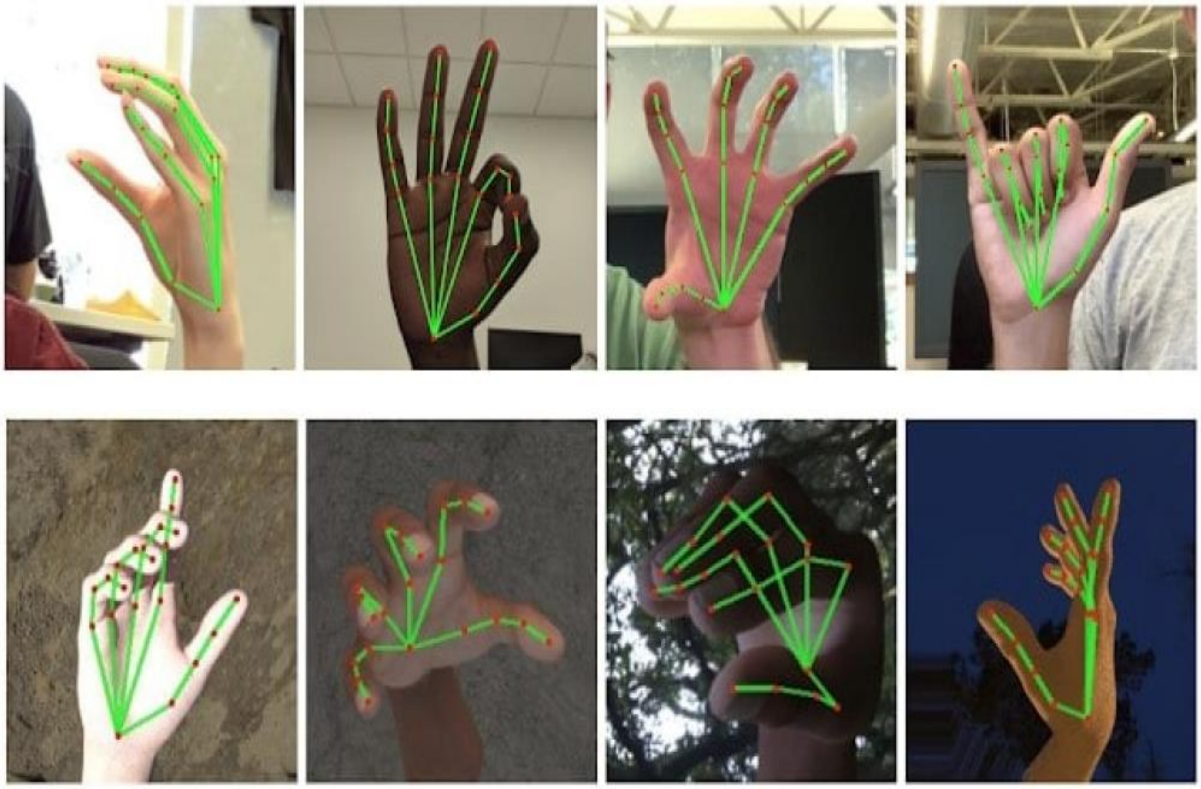
CHAPTER 11

CONCLUSION

In this project a tool is developed which recognises 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 to add multiple images to be uploaded and predict the output for the particular image. For that, user can choose any image from the multiple images and apply the prediction for the respective image for manipulation.

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 useability 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

Image_preprocessing.py:

```
"""Importing Libraries """

import os

from tensorflow.keras.preprocessing.image import ImageDataGenerator

"""Configuring the ImageDataGenerator Class"""

train_data_generator1 =
ImageDataGenerator(rescale=1.0/255,horizontal_flip=True)

test_data_generator1 = ImageDataGenerator(rescale=1.0/255,
horizontal_flip=True)

train_data_generator2 =
ImageDataGenerator(rescale=1.0/255,rotation_range=90)

test_data_generator2 =
ImageDataGenerator(rescale=1.0/255,rotation_range=90)

train_data_generator3 =
ImageDataGenerator(rescale=1.0/255,brightness_range=[0.2,1.0])

test_data_generator3 =
ImageDataGenerator(rescale=1.0/255,brightness_range=[0.2,1.0])

train_data_generator4 =
ImageDataGenerator(rescale=1.0/255,zoom_range=[0.5,1.0])

test_data_generator4 =
ImageDataGenerator(rescale=1.0/255,zoom_range=[0.5,1.0])

"""Applying ImageDataGenerator to test dataset and train dataset"""

trdata1 = train_data_generator1.flow_from_directory(r'D:\College\7th
semester\ibm\Project Development Phase\Sprint 1\Data
collection\Dataset\train',target_size=(64,64),batch_size=(3),color_mode='graysc
ale',class_mode='categorical')
```

```
trdata2 = train_data_generator2.flow_from_directory(r'D:\College\7th semester\ibm\Project Development Phase\Sprint 1\Data collection\Dataset\train',target_size=(64,64),batch_size=(3),color_mode='grayscale',class_mode='categorical')

trdata3 = train_data_generator3.flow_from_directory(r'D:\College\7th semester\ibm\Project Development Phase\Sprint 1\Data collection\Dataset\train',target_size=(64,64),batch_size=(3),color_mode='grayscale',class_mode='categorical')

trdata4 = train_data_generator4.flow_from_directory(r'D:\College\7th semester\ibm\Project Development Phase\Sprint 1\Data collection\Dataset\train',target_size=(64,64),batch_size=(3),color_mode='grayscale',class_mode='categorical')

tsdata1 = test_data_generator1.flow_from_directory(r'D:\College\7th semester\ibm\Project Development Phase\Sprint 1\Data collection\Dataset\test',target_size=(64,64),batch_size=32,class_mode='categorical',color_mode='grayscale')

tsdata2 = test_data_generator2.flow_from_directory(r'D:\College\7th semester\ibm\Project Development Phase\Sprint 1\Data collection\Dataset\test',target_size=(64,64),batch_size=32,class_mode='categorical',color_mode='grayscale')

tsdata3 = test_data_generator3.flow_from_directory(r'D:\College\7th semester\ibm\Project Development Phase\Sprint 1\Data collection\Dataset\test',target_size=(64,64),batch_size=32,class_mode='categorical',color_mode='grayscale')

tsdata4 = test_data_generator4.flow_from_directory(r'D:\College\7th semester\ibm\Project Development Phase\Sprint 1\Data collection\Dataset\test',target_size=(64,64),batch_size=32,class_mode='categorical',color_mode='grayscale')

print(trdata1.class_indices)

print(trdata2.class_indices)

print(trdata3.class_indices)

print(trdata4.class_indices)

print(tsdata1.class_indices)
```

```
print(tsdata2.class_indices)

print(tsdata3.class_indices)

print(tsdata4.class_indices)
```

App.py:

```
from flask import Flask,render_template,request

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

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

import operator

import cv2 # opencv library

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

import numpy as np

from tensorflow.keras.models import load_model #to load our trained model

import os

from werkzeug.utils import secure_filename

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

# Loading the model

model=load_model('gesture.h5')

@app.route('/')# route to display the home page

def home():

    return render_template('home1.html')#rendering the home page

@app.route('/demo',methods=['GET','POST']) # routes to the intro page

def demo():

    return render_template('demo.html')#rendering the intro page
```

```
@app.route('/launch',methods=['GET','POST'])

def launch():

    return render_template("launch.html")

@app.route('/perform',methods=['GET', 'POST'])# route to show the predictions in a web UI

def perform():

    if request.method == 'POST':

        print("inside image")

        file_loader = request.files['image']

        basepath = os.path.dirname(__file__)

        file_path = os.path.join(basepath, 'uploads', secure_filename(file_loader.filename))

        file_loader.save(file_path)

        print(file_path)

        cap = cv2.VideoCapture(0)

        while True:

            _, frame = cap.read() #capturing the video frame values

            # Simulating mirror image

            frame = cv2.flip(frame, 1)

            # Coordinates of the ROI

            x1 = int(0.5*frame.shape[1])

            y1 = 10

            x2 = frame.shape[1]-10

            y2 = int(0.5*frame.shape[1])

            # Drawing the ROI
```

```
        # The increment/decrement by 1 is to compensate for the bounding box
        cv2.rectangle(frame, (x1-1, y1-1), (x2+1, y2+1), (255,0,0) ,1)
        # Extracting the ROI
        roi = frame[y1:y2, x1:x2]
        # Resizing the ROI so it can be fed to the model for prediction
        roi = cv2.resize(roi, (64, 64))
        roi = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
        _, test_image = cv2.threshold(roi, 120, 255, cv2.THRESH_BINARY)
        cv2.imshow("test", test_image)
        # Batch of 1
        result = model.predict(test_image.reshape(1, 64, 64, 1))
        prediction = {'ZERO': result[0][0],
                      'ONE': result[0][1],
                      'TWO': result[0][2],
                      'THREE': result[0][3],
                      'FOUR': result[0][4],
                      'FIVE': result[0][5]}
        # Sorting based on top prediction
        prediction = sorted(prediction.items(), key=operator.itemgetter(1),
reverse=True)
        # Displaying the predictions
        cv2.putText(frame, prediction[0][0], (10, 120),
cv2.FONT_HERSHEY_PLAIN, 1, (0,255,255), 1)
        cv2.imshow("Frame", frame)
        #loading an image
```

```
image1=cv2.imread(file_path)
if prediction[0][0]=='ONE':
    resized = cv2.resize(image1, (200, 200))
    cv2.imshow("Fixed Resizing", resized)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("1"):
        cv2.destroyWindow("Fixed Resizing")
elif prediction[0][0]=='ZERO':
    cv2.rectangle(image1, (480, 170), (650, 420), (0, 0, 255), 2)
    cv2.imshow("Rectangle", image1)
    cv2.waitKey(0)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("0"):
        cv2.destroyWindow("Rectangle")
elif prediction[0][0]=='TWO':
    (h, w, d) = image1.shape
    center = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(center, -45, 1.0)
    rotated = cv2.warpAffine(image1, M, (w, h))
    cv2.imshow("OpenCV Rotation", rotated)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("2"):
        cv2.destroyWindow("OpenCV Rotation")
elif prediction[0][0]=='THREE':
```

```
            blurred = cv2.GaussianBlur(image1, (21, 21), 0)
            cv2.imshow("Blurred", blurred)
            key=cv2.waitKey(3000)
            if (key & 0xFF) == ord("3"):
                cv2.destroyWindow("Blurred")
        elif prediction[0][0]=='FOUR':
            resized = cv2.resize(image1, (400, 400))
            cv2.imshow("Fixed Resizing", resized)
            key=cv2.waitKey(3000)
            if (key & 0xFF) == ord("4"):
                cv2.destroyWindow("Fixed Resizing")
        elif prediction[0][0]=='FIVE':
            gray = cv2.cvtColor(image1, cv2.COLOR_RGB2GRAY)
            cv2.imshow("OpenCV Gray Scale", gray)
            key=cv2.waitKey(3000)
            if (key & 0xFF) == ord("5"):
                cv2.destroyWindow("OpenCV Gray Scale")
        else:
            continue
        interrupt = cv2.waitKey(10)
        if interrupt & 0xFF == 27: # esc key
            break
    cap.release()
    cv2.destroyAllWindows()
```

```
    return render_template("home1.html")
if __name__ == "__main__":
   # running the app
    app.run(debug=True)
```

Demo.html:

```
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>INTRODUCTION/DEMO</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
  </script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js">
  </script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js">
    </script>
<style>
   .topnav {
```

```
    background-color:black;
    overflow: hidden;
}
/* Style the links inside the navigation bar */
.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
/* Change the color of links on hover */
.topnav a:hover {
  background-color: rgb(105, 118, 160);
  color: black;
}
/* Add a color to the active/current link */
.topnav a.active {
  background-color:rgb(105, 118, 160);
  color: white;
}
```

```
/* Right-aligned section inside the top navigation */
.topnav-right {
  float: right;
}
body
{
  background-size:auto;
  background-position:center;
}
h1
{
  text-decoration: underline;
  color: black;
}
</style>
  </head>
 <body>
  <div class="topnav">
    <a class="active" href="{{url_for('demo')}}">Demo</a>
<div class="topnav-right">
      <a href="{{url_for('home')}}">Home</a>
      <a href="{{url_for('launch')}}">Launch</a>
```

```
    </div>
  </div>
<h1 style="color: rgb(193, 207, 207);">
  <center>
  <h1>HAND GESTURE RECOGNITION OF RADIOLOGY IMAGES THROUGH STERILE BROWSING</h1>
       <iframe width="780" height="440" src="https://www.youtube.com/embed/nD621G8u6oc?start=3&loop=1&autoplay=1&mute=1&controls=0">
      </iframe><br/>
    </center>
</div>
</body>
</html>
```

Home1.html:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>HOME</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <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>

    <script src="https://kit.fontawesome.com/bfa13e516e.js"
crossorigin="anonymous"></script>

     <link href="{{ url_for('static', filename='css/style1.css') }}"
rel="stylesheet">

    </script>

</head>

<header class="center">

    <div class="container-fluid">

        <div class="row">

            <nav class="navbar navbar-expand-lg py-3">

                <nav class="navbar navbar-expand-lg navbar-dark shadow-5-
strong"></nav>
```

```
            <a href="#" class="navbar-brand">

              <!-- Logo Image -->

             <img src="{{ url_for('static', filename='images/logoo.png') }}"
width="45" alt=""></a>

              <!-- Logo Text -->

            <a href="https://smartinternz.com/guided-project/a-gesture-based-
tool-for-sterile-browsing-of-radiology-images-cnn-and-open-cv">

              <h2> Gesture-based Tool for Sterile Browsing of Radiology
Images</h2>

           </a>

            </div>

            </div>

          </div>

           </nav>

            <div class="topnav">

              <a class="active" href="{{url_for('home')}}">Home</a>

          <div class="topnav-right">

              <a href="{{url_for('demo')}}">Demo</a>

              <a href="{{url_for('launch')}}">Launch</a>

            </div>

          </div>

          </div>
```

```
    </div>

</header>

<img class="bg" src="{{ url_for('static', filename='images/doc.jpg') }}" width=100% height=100%>

<div class="h1">

    <br>

    <marquee direction="right"> <h1> INTRODUCTION TO HAND GESTURE</h1></marquee>

    </div>

    <h3>Humans are able to recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development .

In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others.

In this project Gesture based Desktop automation ,First the model is trained pre trained on the images of different hand gestures, such as a showing numbers with fingers as 1 ,2,3,4 .

This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with  the Pre-trained model and the gesture is identified.

If the gesture predictes is 1 then images is blurred;2, image is resized;3,image is rotated etc.</h3>



<br>

<br>
```

```
<br>

<section id="about" class="with-medical">

   <div class="container">

      <div class="row">

         <div class="col-lg-6 col-md-12">

            <img class="img-responsive" src="{{ url_for('static',
filename='images/ov.png') }}" alt="" width="500" height="500">

         </div>

         <div class="col-lg-6 col-md-12 txtr">

            <br>

            <h3>OVERVIEW</h3>

            <p>In two brain surgeries at the Neurosurgery OR at the Washington
Hospital Center, procedures were observed by the authors to gain insights about
the use of current technologies and how they affect the quality of the surgeon's
performance.

                  We found that: (a) surgeons kept their focus of attention between
the patient and the surgical point of interest on the touch-screen navigation
system;

                        (b) a short distance between the surgeon and the patient
was maintained during most of the surgery;

                        (c) the surgeon had to move close to the main control
wall to discuss and browse through the patient's MRI images.

The hand gesture control system “Gestix” developed by the authors helped the
doctor to remain in place during the entire operation, without any need to move
to the main control wall since all the commands were performed using hand
gestures.</p>
```

```
        </div>
      </div>
    </div>
</section>
<br>
<br>
<div id="gallery" class="gallery">
    <div class="container">
      <h2><u>RADIOLOGY</h2><u></h2>
  <div class="row">
<div class="gallery_product col-md-4 col-md-4 col-sm-4 col-xs-6">
            <img src="{{ url_for('static', filename='images/c1.jpg') }}"
class="img-responsive" width="300" height="300">
        </div>
        <div class="gallery_product col-md-4 col-md-4 col-sm-4 col-xs-6 ">
            <img src="{{ url_for('static', filename='images/c2.jpg') }}"
class="img-responsive" width="300" height="300">
        </div>
        <div class="gallery_product col-md-4 col-md-4 col-sm-4 col-xs-6 ">
            <img src="{{ url_for('static', filename='images/c3.jpg') }}"
class="img-responsive" width="300" height="300">
        </div>
        </div>
```

```
    </div>
</div>
</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=1">
      <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>MANIPULATE</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">

    <script src="https://kit.fontawesome.com/bfa13e516e.js"
crossorigin="anonymous"></script>

 <style>

 .bar

 {

 margin: 0px;

 padding:20px;

 background-color:black;

 opacity:0.6;

 color:black;

 font-family:'Roboto',sans-serif;

 font-style: italic;

 border-radius:20px;

 font-size:25px;

 }

 a

 {

 color:rgb(181, 228, 236);

 float:right;

 font-weight: bold;

 text-decoration:none;
```

```
font-style:normal;
padding-right:20px;
font-size: 30px;
}
div1{
  text-align: center;
  width: 650spx;
  height: 800px;
  padding: 180px;

  margin: 10px;
  position: absolute;
}
body
{
  background: url("{{ url_for('static', filename='images/fin.jpg') }}") no-repeat center center fixed ;
  -webkit-background-size: cover;
-moz-background-size: cover;
-o-background-size: cover;
background-size: cover;
}
```

```
    .topnav {
  background-color:black;
  overflow: hidden;
}
/* Style the links inside the navigation bar */
.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
/* Change the color of links on hover */
.topnav a:hover {
  background-color: rgb(105, 118, 160);
  color: black;
}
/* Add a color to the active/current link */
.topnav a.active {
  background-color:rgb(105, 118, 160);
  color: white;
```

```
}
/* Right-aligned section inside the top navigation */
.topnav-right {
  float: right;
}
  .upload{
    border: 1px solid black;
    border-radius: 50%;
    padding: 20px;
    background-color: white;
  }
  .button {
  background-color: #091425;
  border: none;
  color: black;
  padding: 15px 32px;
  text-align: center;
  text-decoration: 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);
}
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  display: inline-block;
  margin-bottom:18px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}
form{
  margin-left: 400px;
  margin-right: 400px;
}
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 {
  text-align: 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;
```

```
}
.contents{
  margin-left: 120px;
}
.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: background-color 0.6s ease;
}
.link {
  font-size: 17px;
}
/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 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 class="side" onclick="changeicon(this)"><div class="bar1"></div>
    <div class="bar2"></div>
    <div class="bar3"></div></div>
  </div>
  <br>
  <div class="topnav">
    <a class="active" href="{{url_for('launch')}}">Launch</a>
<div class="topnav-right">
      <a href="{{url_for('home')}}">Home</a>
      <a href="{{url_for('demo')}}">Demo</a>
    </div>
  </div>
    <div1 class="contents"><h1><font color="Black" size="6" font-family="Roboto">Hand Gesture Recognition</h1><br>
    <h4><i><font color="Black" size="4" fonr-family="sans-serif"></i><u>Provide an image for which you want to perform various operations</u></h4>
```

```
  <br>
    <div>
      <h4>Upload Image Here</h4>
    <form action = "http://localhost:5000/" id="upload-file" method="post" enctype="multipart/form-data">
    <br>
    <div class="upload">
    <label><i class="fa fa-upload" style="font-size: 50px; "aria-hidden="true"><input type="file" style="display:none;" name="image" id="imageUpload" accept=".png, .jpg, .jpeg,.pdf"></i></label>
    </div><br><br>
    </form>

    <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>
    </div>
    </div1>
        <footer>
    <script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
  </footer>
  </body>
  </html>
```

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 20px;
    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); }
}
```

Style1.css:

```
body
{
background-size:cover;
background-position-y:1270px;
}
header
{
    background:rgba(255,255,255,0.6);
}
h1
{
    text-align: center;
    text-decoration: underline;
    text-shadow: 2px;
}
h3
```

```
{
    text-align:left;
    font-style: normal;
    font-weight: 100;
    text-align-last: left;
}
#p1
{
    padding: 50px 50px;
  border: 2px solid #162b78;
  background-position: left;
}

    #about{
  border: 3px dotted;
border-color: rgb(0, 0, 0);

}
.topnav {
    background-color:black;
    overflow: hidden;
}
/* Style the links inside the navigation bar */
.topnav a {
```

```
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
/* Change the color of links on hover */
.topnav a:hover {
  background-color: rgb(105, 118, 160);
  color: black;
}
/* Add a color to the active/current link */
.topnav a.active {
  background-color:rgb(105, 118, 160);
  color: white;
}
/* Right-aligned section inside the top navigation */
.topnav-right {
  float: right;
}
```

Model_building.py:

```
"""Model Building.ipynb
"""Import the libraries"""
```

```
import numpy as np

import os

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense,Flatten,Dropout

from tensorflow.keras.layers import Convolution2D,MaxPooling2D

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.preprocessing import image

from tensorflow.keras.models import load_model

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

"""Augment the data"""

train = ImageDataGenerator(rescale =
1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)

test = ImageDataGenerator(rescale = 1./255)

"""Loading and augmentation of given data"""

A_train = train.flow_from_directory(r'D:\College\7th semester\ibm\Final
Deliverables\Project\Dataset\train', target_size=(64,64),
color_mode='grayscale',batch_size=3, class_mode='categorical')

A_test = test.flow_from_directory(r'D:\College\7th semester\ibm\Final
Deliverables\Project\Dataset\test', target_size=(64,64),
color_mode='grayscale',batch_size=3, class_mode='categorical')

print(A_train.class_indices)

print(A_test.class_indices)

"""Import Keras library"""

model = Sequential()
```

```
"""Add 1st Convolution Layer and Pooling  layer"""

model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

"""Add 2nd Convolution Layer and Pooling layer"""

model.add(Convolution2D(32,(3,3),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

"""Add Flatten layer"""

model.add(Flatten())

"""Add dense layers"""

model.add(Dense(units=512,activation='relu'))

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

print(model.summary())

"""Compile the model"""

model.compile(metrics=['accuracy'],loss='categorical_crossentropy',optimizer='
adam')

"""Train the model"""

model.fit(A_train,steps_per_epoch =
594/3,epochs=25,validation_data=A_test,validation_steps=len(A_test))

"""Save the model"""

model.save('gesture.h5')

json_model = model.to_json()

with open("model-gesture.json","w") as json_file:

  json_file.write(json_model)

"""Test the model"""

test_model = load_model('gesture.h5')
```

```
img_path=r"D:\College\7th semester\ibm\Final Deliverables\Project\Model Building\test_image.jpg"

img = mpimg.imread(img_path)

imgplot = plt.imshow(img)

plt.show()

imgload = image.load_img(img_path,color_mode='grayscale',target_size=(64,64))

res = image.img_to_array(imgload)

print(res.shape)

print(type(res))

res = np.expand_dims(res,axis=0)

print(res.shape)

"""Predict the result"""

pred_res = np.argmax(test_model.predict(res),axis=-1)

print(pred_res)

index = ['0','1','2','3','4','5']

final_res = str(index[pred_res[0]])

print(final_res)
```

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: '/perform',
            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!');
            },
        });
    });
});
```

Github Link:

https://github.com/IBM-EPBL/IBM-Project-43342-1660716108

Youtube Demonstration Video:

https://youtu.be/K9eYGN4LOA0