

## PROJECT REPORT

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

Team ID : **PNT2022TMID18569**

Team Leader – **Sujitha AK** – 1919102163

Team Member – **Swetha T** - 1919102170

Team Member – **Vakinidevi N** – 1919102174

Team Member – **Swetha S** - 1919102169

# **CONTENT**

## **1. INTRODUCTION**

1. Project Overview

## **2. LITERATURE SURVEY**

1. Existing problem
2. References
3. Problem Statement Definition

## **3. IDEATION & PROPOSED SOLUTION**

1. Empathy Map Canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. Problem Solution fit

## **4. REQUIREMENT ANALYSIS**

1. Functional requirement
2. Non-Functional requirements

## **5. PROJECT DESIGN**

1. Data Flow Diagrams
2. Solution & Technical Architecture
3. User Stories

## **6. PROJECT PLANNING & SCHEDULING**

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule
3. Reports from JIRA

## **7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

1. Feature 1
2. Feature 2

## **8. TESTING**

1. Test Cases
2. User Acceptance Testing

## **9. RESULTS**

1. Performance Metrics
  2. Output
  10. **ADVANTAGES & DISADVANTAGES**
  11. **CONCLUSION**
  12. **FUTURE SCOPE**
  13. **APPENDIX**
- 
1. Source Code
  2. GitHub & Project Demo Link

## **1.INTRODUCTION :**

### **1.1 Project Overview**

- The main aim of the project is to build a model that displays the images based on hand gestures. Gestures originate from any bodily motion or state but commonly originate from the face or hand. Gesture Recognition can be considered as a way for a computer to understand human body language. This has minimized the need for text interfaces and GUIs (Graphical User Interface). A gesture is an action that has to be seen by someone else and has to convey some piece of information. The gesture is usually considered as a movement of part of the body, esp. a hand or the head, to express an idea or meaning. Gesture recognition technologies are much younger in the world of today.
- Gestures are useful for computer interaction since they are the most primary and expressive forms of human communication. Gesture interfaces for gaming based on hand/body gesture technology must be designed to achieve social and commercial success. No single method for automatic hand gesture recognition is suitable for every application; each gesture-recognition algorithm depends on user cultural background, application domain, and environment.

## **2.LITERATURE SURVEY :**

### **2.1 Existing Problem**

- A major challenge involved is to provide Doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work.
- However the use of computer keyboards and mouse by doctors in intensive care unit(ICU) is a common mean for spreading infections.
- We suggest the use of hand gestures in medical field as an alternative to the existing interface techniques by offering maximum level of sterility.

### **2.2 References**

- Robust Part-Based Hand Gesture Recognition Using Kinect SensorZhou Ren, Junsong Yuan, Member, IEEE, Jingjing Meng, Member, IEEE, and Zhengyou Zhang, Fellow, IEEE, 15, AUGUST 2013.
- A Fast Gesture Recognition Scheme for Real-Time HumanMachine Interaction Systems . Ching-Hao Lai\* Smart Network System Institute for Information Industry Taipei City, Taiwan , 2010.
- Intension, Context and Gesture Recognition for Sterile MRI Navigation in the Operating Room by Agency for Healthcare Research and Quality (AHRQ)
- Hand Gestures Recognition Using Radar Sensors for Human-ComputerInteraction Supported by the Bio ad Medical Technology Development Program of the National Research Foundation(NRF)
- A Preliminary Study of Kinect-Based Real-Time Hand Gesture Interaction Systems for Touchless Visualizations of Hepatic Structures in Surgery by Medical Imaging and Information Sciences, Jiaqing LIU, Tomoko Tateyama.

- Vision Based Hand Gesture Recognition by World Academy of Science, Engineering and Technology, Pragati Garg, Naveen Aggarwal, Sanjeev Sofat

## 2.3 Problem Statement Definition

The solution is to develop AI-powered gesture based tool.

By giving the image of the hand gesture as the input to the application, it will display the required images.

By training the model with various inputs, image processing can be improved as well as the accuracy of the result.

## 3. IDEATION & PROPOSED SOLUTION :

### 3.1 Empathy Map Canvas




## 3.2 Ideation & 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.

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

**Step-2:** Brainstorm, Idea Listing and Grouping.

**Step-3:** Idea Prioritization.



### Brainstorm & idea prioritization

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

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

bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

**Before you collaborate**

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

10 minutes

- Team gathering**  
Define who should participate in the session invite. Share relevant information or preview
- Set the goal**  
Think about the problem you'll be focusing the brainstorming session.
- Learn how to use the facilitation tools**  
Use the Facilitation Superpowers to run a h productive session.

Open article →

**1 Define your problem statement**

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

5 minutes

Problem

A Gesture based tool for sterile browsing of radiology images.

Problem Statement

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 most appropriate among others.

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

20 minutes

**2 Brainstorm**

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

10 minutes

Sujitha AK	Swetha T
Mainly used in operation room	Support medical imaging manipulation
Providing fast response time	Navigation are translated into commands
	Helps to achieve non-verbal conversational modalities
	Offering major advantage of sterility

Vakinidevi N	Swetha S
Substitute of verbal based communication	Sensors read and interpret hand movements
Could replace use of keyboard	Algorithms were used
	Sensors read and interpret hand movements

Uses integrated web cam to capture video frame	Video frame compared with pre trained model	Image pre-processing
Motion of body contains information	Interpreting human interactions via mathematical algorithms	Gesture as input to control devices
Adding CNN and dense	Gestures do not have universal	Gesture includes face, hands or

**Importance**

If each of these tasks could get done, which would be most difficult or most valuable to your problem domain?

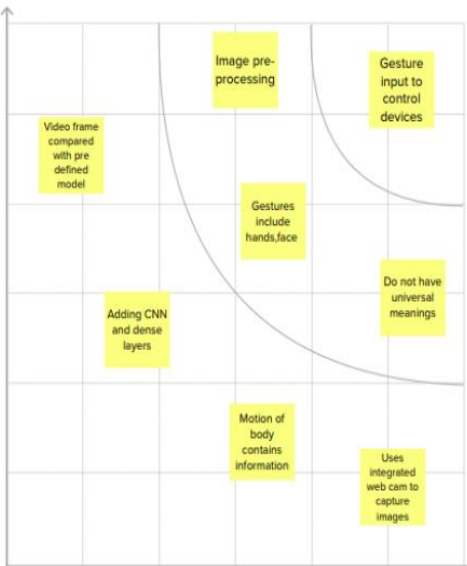


Image pre-processing

Gesture input to control devices

Video frame compared with pre defined model

Gestures include hands, face

Do not have universal meanings

Adding CNN and dense layers

Motion of body contains information

Uses integrated web cam to capture images

### 3.3 Proposed Solution

- The interaction with interventional imaging systems within a sterile environment is a challenging task for physicians. Direct physician–machine interaction during an intervention is rather limited because of sterility and workspace restrictions.
- We propose a method of gesture-controlled projection display that enables a direct and natural physician–machine interaction during computed tomography (CT)-based interventions.
- Therefore, a graphical user interface is projected on a radiation shield located in front of the physician. Hand gestures in front of this display are captured and classified using a leap motion controller.
- We propose a gesture set to control basic functions of intervention software such as gestures for 2D image exploration, 3D object manipulation and selection. Our methods were evaluated in a clinically oriented user study with 12 participants.
- The results of the performed user study confirm that the display and the underlying interaction concept are accepted by clinical users. The recognition of the gestures is robust, although there is potential for improvements.
- The gesture training times are less than 10 min, but vary heavily between the participants of the study. The developed gestures are connected logically to the intervention software and intuitive to use.
- The proposed gesture-controlled projection display counters current thinking, namely it gives the radiologist complete control of the intervention software. It opens new possibilities for direct physician–machine interaction interventions most importantly during surgeries.

## 3.4 Problem Solution Fit

Define CS, fit into CC	<div>1. CUSTOMER SEGMENT(S)<div>CS</div><div>Who is your customer?</div><div>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 "Gestor," 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. "Gestor" 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.</div></div>	<div>6. CUSTOMER<div>CC</div><div>What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.</div><div>(i) ease of use—the system allows the surgeon to use his/her hands, their natural work tool</div><div>(ii) rapid reaction—nonverbal instructions by hand gesture commands are intuitive and fast (In practice, the "Gestor" system can process images and track hands at a frame-rate of 150 Hz, thus, responding to the surgeon's gesture commands in real-time)</div><div>(iii) an unencumbered interface—the proposed system does not require the surgeon to attach a microphone, use head-mounted (body-contact) sensing devices or to use foot pedals</div><div>(iv) distance control—the hand gestures can be performed up to 5 meters from the camera and still be recognized accurately. The results of two usability tests (contextual and individual interviews) .</div></div>	<div>5. AVAILABLE SOLUTIONS<div>AS</div><div>Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros &amp; cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking</div><div>In this work, the user is allowed to interact with the medical image and 3D visualized model using a touch-less based interaction technique. This method is done by recognizing the gestures shown by the user. Gestures recognition process become much easier by using inexpensive sensor. Application is developed to recognize the gestures and simulate inputs corresponding to gestures. Each gesture has its own operation on the medical image or 3D volumetric visualization.</div></div>	Explore AS, differentiate
	<div>2. JOBS-TO-BE-DONE / PROBLEMS<div>J&amp;P</div><div>Which jobs-to-be-done (or problems) do you address for your customers?There could be more than one; explore different sides.</div><div>According to [4] human-computer interaction is fundamental to the discipline of information technology. Any computing system is said to be interactive when it involves one or more interface that allows users to give commands and get results. Development using the graphical user interface, which gives the users from varying levels of knowledge about computers, allows them to understand how to use the application.</div></div>	<div>9. PROBLEM ROOT CAUSE<div>RC</div><div>What is the real reason that this problem exists? What is the back story behind the need to do this job?</div><div>i.e. customers have to do it because of the change in regulations.</div><div>A tracker to track and recognize independent hand posture and trajectory and send the information to the host application. Improvement is made in [10] by replacing the tracking system with a Wii mote to allow exploration for medical data at a distance. The idea from these projects about the gesture control interface are very convincing but these solutions are unsuitable to be used in the medical field especially in the operating rooms where non-stabilizable devices are not permitted to avoid contamination of the patient, the OR and the surgeon. There are also vision-based interfaces have been proposed to be used in the operation room. A system named Gestor is presented in [11] which is a video-based hand gesture recognition system. It allows user to navigate and manipulate magnetic resonance images (MRI). The problem with the system is vision-based interfaces need to contend with other problems since the vision can be affected by room lighting, user movement and the background of the user.</div></div>	<div>7. BEHAVIOUR<div>BE</div><div>What does your customer do to address the problem and get the job done?</div><div>i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)</div><div>Medical image contains many slices that can be process and visualize in 3D volumetric model. The medical visualization application allows users to change the image slice so that they can study the image from different position. Changing image slice can be done by swiping one hand in up or down direction to change the image slice. This gesture is only valid to be used on 2D medical image.</div></div>	Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC	<div>3. TRIGGERS<div>TR</div><div>What triggers customers to act? i.e. seeing their neighbor installing solar panels, reading about a more efficient solution in the news.</div><div>The available gestures in the application used to control and manipulate the medical image and visualization model. Rotation can be done using 1 hand at the front, pointing at the start position and drag to rotate it based on the needs to rotate the visualize model. We can see that action signifies holding and turning the object.</div></div>	<div>10. YOUR SOLUTION<div>SL</div><div>If you are working on an existing business, write down your current solution first,fill in the canvas, and check how much it fits reality.</div><div>If you are working on a new business proposition, then keep it blank until you fill inthe canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behavior.</div><div>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.</div></div>	<div>8. CHANNELS of BEHAVIOUR<div>CH</div><div>8.1 ONLINE</div><div>What kind of actions do customers take online? Extract online channels from #7</div><div>The gesture that provided in the application that can be used in the application. There are also gestures that can be used to do operation that not related in navigate and manipulate the visualization. Each gesture has its own operation to avoid confusion.</div><div>8.2 OFFLINE</div><div>What kind of actions do customers take offline? Extract offline channels from #7and use them for customer development.</div><div>Kinect sensor and the medical visualization application so that the visualized model is ready to be used with the application. The interaction flow start with hand and arm gestures from user. Kinect will track the user movement and send the information to the application for the gesture recognition process.</div></div>	Extract online & offline CH of BE
Identify strong TR & EM	<div>4. EMOTIONS: BEFORE / AFTER<div>EM</div><div>How do customers feel when they face a problem or a job and afterwards?</div><div>i.e. lost, insecure &gt; confident, in control - use it in your communication strategy &amp; design.</div><div>The essential hardware required for this application to run is the Kinect itself as the camera used to track the user. There are many cameras that equipped with depth sensor by Kinect is among the best in term of price and performance. The availability of Kinect also is the one of the factor because it is easily to find all over the world.</div></div>			



## 4.REQUIREMENT ANALYSIS :

### 4.1 Functional Requirements

#### Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Launching the model	Launch the trained CNN model from the cloud
FR-2	Capturing the images	After capturing the images in camera we have to upload the images in the system
FR-3	Performing gestures	After classifying, identify the correct image by the gesture and it should perform the operation
FR-4	Model rendering	After capturing the image the algorithm will start its processing task
FR-5	Sterile browsing	The sterile browsing can be performed after identifying the gestures
FR-6	Visibility of images	After completing all the processes,a user can be able to see the images

### 4.2 Non - Functional Requirements

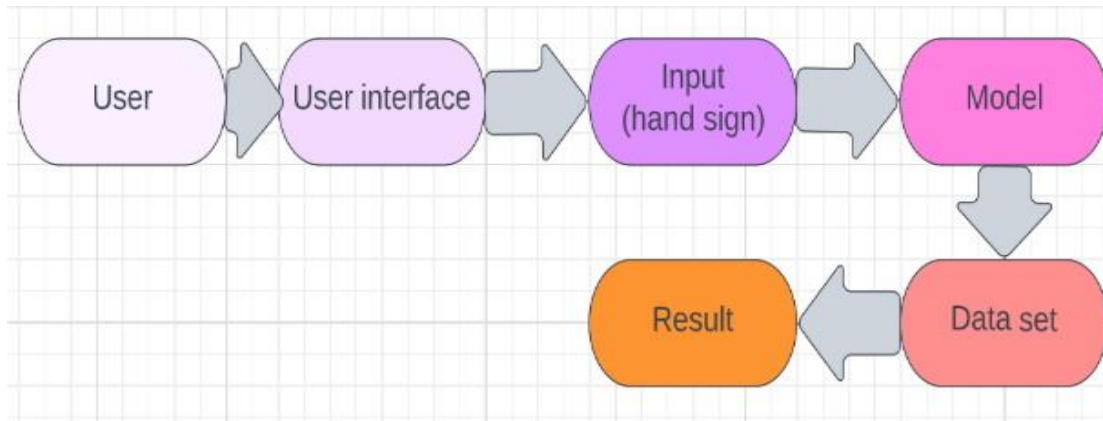
#### Non-functional Requirements:

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

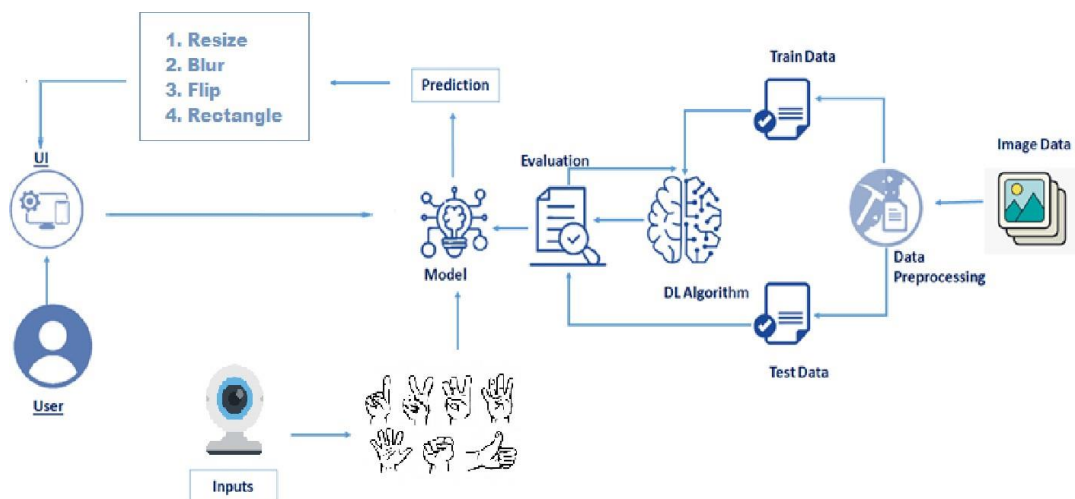
FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	This system helps to have the control over images without having direct contact with system which avoids the harmful rays and is ease of use
NFR-2	<b>Security</b>	This system is protected and only authorized users can access it
NFR-3	<b>Reliability</b>	After installing the application,the system will predict the gesture and performs sterile browsing
NFR-4	<b>Performance</b>	The system responds to a user in seconds and the hardware and software works well
NFR-5	<b>Availability</b>	It is accessible by authorised user from anywhere at any time whenever there is an emergency
NFR-6	<b>Scalability</b>	This system allows more number of users at a time and there is no loss can be identified

## 5.PROJECT DESIGN :

## 5.1 Data Flow Diagrams



## 5.2 Solution & Technical Architecture



## 5.3 User Stories

### User Stories

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

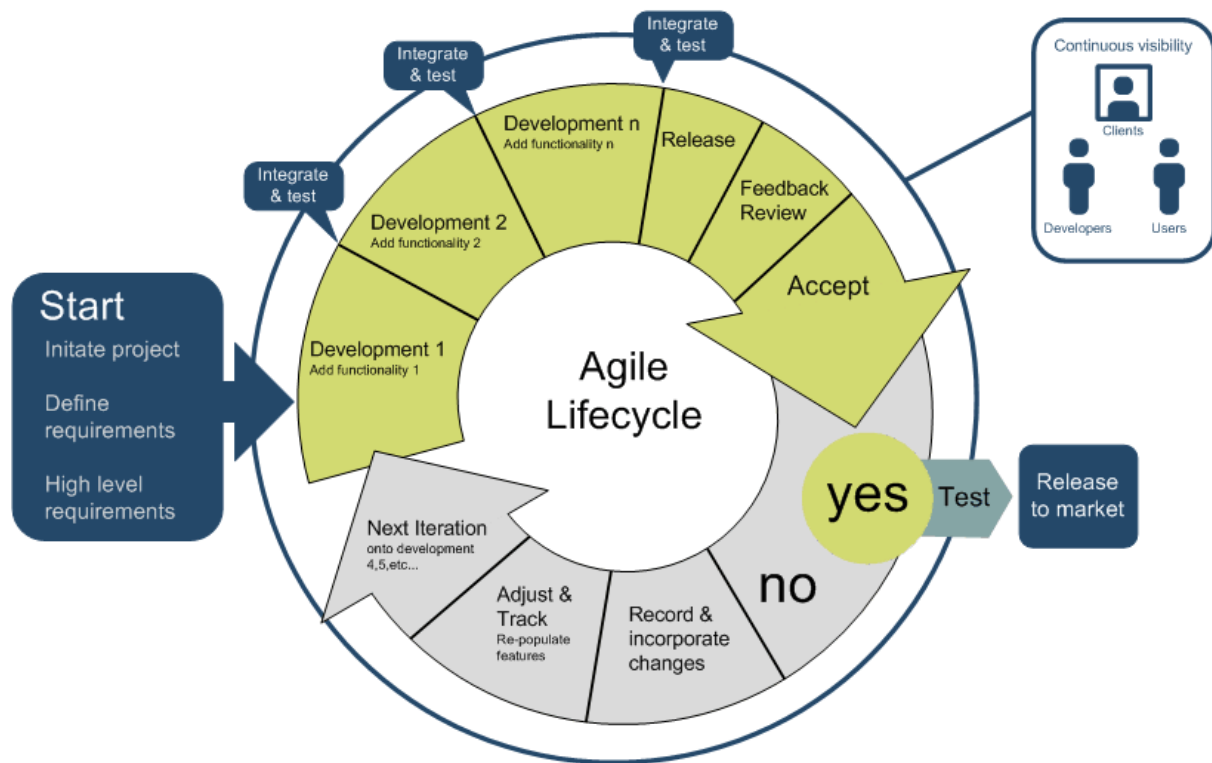
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can upload an image for performing the action.	I can upload image	High	Sprint-3
		USN-2	As a user, I can show my hand sign in front of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

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

To accomplish the above task, you must complete the below activities and tasks:

1. Collect the Image Data.
2. Preprocess the collected images.
3. Train the model.
4. Test the model.
5. Model is generated (gesture.h5).
6. Application building using HTML & CSS.
7. Form for uploading the image for prediction.
8. Python flask for connecting Model and webpages.



## 6.2 Sprint Delivery Schedule

### Product Backlog, Sprint Schedule, and Estimation

#### Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

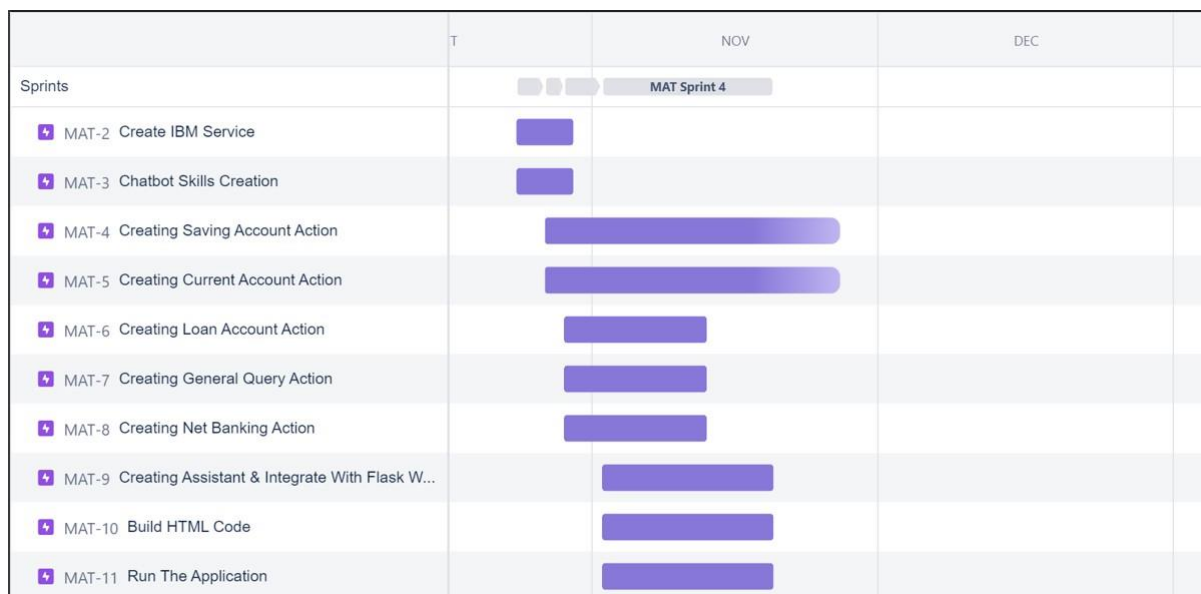
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Image preprocessing	USN-1	As a user, I can predict the image	2	High	Sujitha AK Swetha T Vakinidevi N Swetha S
Sprint-1	Model building	USN-2	As a user, I can predict the model building application	1	High	Sujitha AK Swetha T Vakinidevi N Swetha S
Sprint-2	Launching Software	USN-3	As a user, I can launch the development software	2	Low	Sujitha AK Swetha T Vakinidevi N Swetha S
Sprint-3	Launching Camera	USN-4	As a user, I can open the camera from the software to perform gesture	2	Medium	Sujitha AK Swetha T Vakinidevi N Swetha S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Upload image from local system	USN-5	As a user, I can upload a image to the software from the local system	1	Low	Sujitha AK Swetha T Vakinidevi N Swetha S
Sprint-4	Perform gesture	USN-6	As a user, I can perform various gesture with respect to system specification for processing	2	Medium	Sujitha AK Swetha T Vakinidevi N Swetha S
Sprint-5	output	USN-7	As a user, I can see the sterile browsers image with respect to the gesture performed, display on the screen	2	High	Sujitha AK Swetha T Vakinidevi N Swetha S

#### Project Tracker, Velocity & Burndown Chart: (4 Marks)

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

## 6.3 Reports from JIRA



## 7.CODING & SOLUTIONING :

### 7.1 Feature 1

#### 1:IMAGE PREPROCESSING:

Import the ImagesDataGenerator Library:

```
from keras.preprocessing.image import ImageDataGenerator
```

## Configure ImageDataGenerator Functionality To Trainset And Testset:

```
#Image Data Agumentation
```

```
train_datagen = ImageDataGenerator (rescale=1./255,  
                                   shear_range=0.2, zoom_range=0.2,  
                                   horizontal_flip=True)  
test_datagen=ImageDataGenerator(rescale=1./255)
```

## Apply ImageDataGenerator Functionality To Trainset And Testset

```
#Loading our data and performing data agumentation
```

```
x_train = train_datagen.flow_from_directory(r'C:\Users\Boopathi\Documents\IBM Project\Dataset\training_set',target_size=(64, 64),  
                                           batch_size=32,  
                                           color_mode='grayscale',  
                                           class_mode='categorical')  
x_test = test_datagen.flow_from_directory(r'C:\Users\Boopathi\Documents\IBM Project\Dataset\test_set',  
                                         target_size=(64, 64), batch_size=32,  
                                         color_mode='grayscale',  
                                         class_mode='categorical')
```

```
Found 594 images belonging to 6 classes.  
Found 30 images belonging to 6 classes.
```

## 2:MODEL BULIDING:

### Importing The Model Building Libraries

```
import numpy as np  
import tensorflow as tf  
from tensorflow.keras.models import Sequential  
from tensorflow.keras import layers  
from tensorflow.keras.layers import Dense, Flatten  
from tensorflow.keras.layers import Conv2D, MaxPooling2D  
from keras.preprocessing.image import ImageDataGenerator
```

### Initializing The Mode

```
model=Sequential()
```

## Adding CNN Layers

```
# First convolution Layer and pooling
model.add(Conv2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
# Second convolution Layer and pooling
model.add(Conv2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
model.add(MaxPooling2D(pool_size=(2, 2)))
# Flattening the Layers
model.add(Flatten())
```

## Adding Dense Layers

```
model.add(Dense (units=128, activation='relu'))
model.add(Dense (units=6, activation='softmax'))
```

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 62, 62, 32)	320
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 128)	802944
dense_1 (Dense)	(None, 6)	774
=====		
Total params: 813,286		
Trainable params: 813,286		
Non-trainable params: 0		





## Configure The Learning Process

### # Compiling the model

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

## Training Model

```
model.fit_generator(train_generator,
                    steps_per_epoch=len(train_data),
                    epochs=25,
                    validation_data=validation_data,
                    validation_steps=len(validation_data))
```

C:\Users\Bosch\Anaconda3\envs\python36\python.exe: C:\Users\Bosch\AppData\Local\Temp\python36\python36.exe: UserWarning: "model.fit\_generator" is deprecated and will be removed in a future version. Please use "model.fit", which supports generators.

```
model.fit_generator(train_generator,
                    steps_per_epoch=len(train_data),
                    epochs=25,
                    validation_data=validation_data,
                    validation_steps=len(validation_data))
```

Epoch 1/25  
10/10 [=====] - 2s 80ms/step - loss: 1.7468 - accuracy: 0.5828 - val\_loss: 1.5119 - val\_accuracy: 0.6003  
Epoch 2/25  
10/10 [=====] - 2s 80ms/step - loss: 1.1700 - accuracy: 0.6787 - val\_loss: 0.8810 - val\_accuracy: 0.6008  
Epoch 3/25  
10/10 [=====] - 1s 79ms/step - loss: 0.8158 - accuracy: 0.8084 - val\_loss: 0.5489 - val\_accuracy: 0.7103  
Epoch 4/25  
10/10 [=====] - 2s 79ms/step - loss: 0.6171 - accuracy: 0.7920 - val\_loss: 0.5905 - val\_accuracy: 0.6587  
Epoch 5/25  
10/10 [=====] - 2s 80ms/step - loss: 0.5723 - accuracy: 0.7744 - val\_loss: 0.5410 - val\_accuracy: 0.6223  
Epoch 6/25  
10/10 [=====] - 2s 80ms/step - loss: 0.4474 - accuracy: 0.8055 - val\_loss: 0.4645 - val\_accuracy: 0.6053  
Epoch 7/25  
10/10 [=====] - 2s 80ms/step - loss: 0.3821 - accuracy: 0.8009 - val\_loss: 0.4051 - val\_accuracy: 0.6057  
Epoch 8/25  
10/10 [=====] - 2s 80ms/step - loss: 0.3120 - accuracy: 0.8676 - val\_loss: 0.2682 - val\_accuracy: 0.6908  
Epoch 9/25  
10/10 [=====] - 2s 79ms/step - loss: 0.2847 - accuracy: 0.9158 - val\_loss: 0.4042 - val\_accuracy: 0.6333  
Epoch 10/25  
10/10 [=====] - 1s 79ms/step - loss: 0.1197 - accuracy: 0.9035 - val\_loss: 0.5847 - val\_accuracy: 0.7008  
Epoch 11/25  
10/10 [=====] - 2s 80ms/step - loss: 0.1224 - accuracy: 0.9158 - val\_loss: 0.5814 - val\_accuracy: 0.6908  
Epoch 12/25  
10/10 [=====] - 1s 79ms/step - loss: 0.2142 - accuracy: 0.9026 - val\_loss: 0.2207 - val\_accuracy: 0.6957  
Epoch 13/25  
10/10 [=====] - 1s 79ms/step - loss: 0.1971 - accuracy: 0.9030 - val\_loss: 0.5154 - val\_accuracy: 0.6003  
Epoch 14/25  
10/10 [=====] - 2s 80ms/step - loss: 0.1153 - accuracy: 0.9461 - val\_loss: 0.2882 - val\_accuracy: 0.6923  
Epoch 15/25  
10/10 [=====] - 2s 80ms/step - loss: 0.1448 - accuracy: 0.9513 - val\_loss: 0.2847 - val\_accuracy: 0.6957  
Epoch 16/25  
10/10 [=====] - 2s 80ms/step - loss: 0.1780 - accuracy: 0.9451 - val\_loss: 0.2278 - val\_accuracy: 0.6903  
Epoch 17/25  
10/10 [=====] - 2s 80ms/step - loss: 0.1187 - accuracy: 0.9527 - val\_loss: 0.2422 - val\_accuracy: 0.6957  
Epoch 18/25  
10/10 [=====] - 2s 79ms/step - loss: 0.1133 - accuracy: 0.9588 - val\_loss: 0.2752 - val\_accuracy: 0.6923  
Epoch 19/25  
10/10 [=====] - 2s 79ms/step - loss: 0.1118 - accuracy: 0.9579 - val\_loss: 0.3089 - val\_accuracy: 0.6903  
Epoch 20/25  
10/10 [=====] - 2s 79ms/step - loss: 0.1184 - accuracy: 0.9513 - val\_loss: 0.2870 - val\_accuracy: 0.6923  
Epoch 21/25  
10/10 [=====] - 1s 79ms/step - loss: 0.8695 - accuracy: 0.8724 - val\_loss: 0.4051 - val\_accuracy: 0.6057  
Epoch 22/25  
10/10 [=====] - 2s 79ms/step - loss: 0.8981 - accuracy: 0.8587 - val\_loss: 0.2052 - val\_accuracy: 0.6923  
Epoch 23/25  
10/10 [=====] - 2s 79ms/step - loss: 0.1024 - accuracy: 0.9588 - val\_loss: 0.2782 - val\_accuracy: 0.6923  
Epoch 24/25  
10/10 [=====] - 2s 79ms/step - loss: 0.8777 - accuracy: 0.8747 - val\_loss: 0.4042 - val\_accuracy: 0.6923  
Epoch 25/25  
10/10 [=====] - 2s 79ms/step - loss: 0.8674 - accuracy: 0.8587 - val\_loss: 0.2249 - val\_accuracy: 0.6923

Complete training history at 8/12/2016 12:00

## Save the Model

```
model.save('gesture.h5')
```

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

## Test the Model

## Test the Model

```
import tensorflow as tf
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model = load_model("Gesture.h5")
path= r"C:\Users\Boopathi\Documents\IBM Project\Dataset\test_set\2\1.jpg"
```

```
img = image.load_img(path,
                      color_mode='grayscale',
                      target_size= (64,64))
x = image.img_to_array(img)#image to array
x.shape
```

```
(64, 64, 1)
```

```
type(x)
```

```
numpy.ndarray
```

```
x = np.expand_dims(x,axis=0)
```

```
x.shape
```

```
(1, 64, 64, 1)
```

```
pred=(model.predict(x) > 0.5).astype("int64")
pred
```

```
1/1 [=====] - 0s 141ms/step
```

```
array([[0, 0, 1, 0, 0, 0]], dtype=int64)
```

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

```
index[np.argmax(pred[0])]
```

```
'2'
```

## 7.2 Feature 2

- Let us build flask file 'app.py' which is a web framework written in python for server-side scripting. Let's see step by step procedure for building the backend application.
- App starts running when "\_\_name\_\_" constructor is called in main.
- render\_template is used to return html file.
- "GET" method is used to take input from the user.
- "POST" method is used to display the output to the user.

## 1:Importing Libraries

```
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 os
from tensorflow.keras.models import load_model # to load our trained model import os
from werkzeug.utils import secure_filename
```

## 2:Creating our flask application and loading our model

```
app = Flask(__name__, template_folder="templates") # initializing a flask app
# Loading the model
model = load_model('gesture.h5')
print("Loaded model from disk")
```

## 3:Routing to the html page

```
@app.route('/') # route to display the home page
def home():
    return render_template('home.html') # rendering the home page

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

@app.route('/image1', methods=['GET', 'POST']) # routes to the index html
def index6():
    return render_template("index6.html")

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

And the predict route is used for prediction and it contains all the codes which are used for predicting our results.

- Firstly, inside launch function we are having the following things:
  - Getting our input and storing it
  - Grab the frames from the web cam.
  - Creating ROI
  - Predicting our results
  - Showcase the results with the help of opencv
  - Finally run the application
- Getting our input and storing it

Once the predict route is called, we will check whether the method is POST or not if is POST then we will request the image files and with the help of os function we will be storing the image in the uploads folder in our local system.

```
if request.method == 'POST':
    print("inside image")
    f = request.files['image']

    basePath = os.path.dirname(__file__)
    file_path = os.path.join(basePath, 'uploads', secure_filename(f.filename))
    f.save(file_path)
    print(file_path)
```

- Grab the frames from the web cam

Now when we run the code a web cam will be opening to take the gesture input so we will be capturing the frames of the gesture for predicting our results.

```
cap = cv2.VideoCapture(0)
while True:
    _, frame = cap.read() # capturing the video frame values
    # Simulating mirror image
    frame = cv2.flip(frame, 1)
```

- Creating ROI

A region of interest (ROI) is a portion of an image that you want to filter or operate on in some way. The toolbox supports a set of ROI objects that you can use to create ROIs of many shapes, such as circles, ellipses, polygons, rectangles, and hand-drawn shapes. ... A common use of an ROI is to create a binary mask image.

So, we will be creating a ROI to mask our gesture.

```
# 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)
```

- Predicting our results

After placing the ROI and getting the frames from the web cam now its time to predict the gesture result using the model which we trained and stored it into a variable for the further operations.

- Showcase the results with the help of opencv

Finally according to the result predicted with our model we will be performing certain operations like resize, blur , rotate etc.

```
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.destroyAllWindows("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.destroyAllWindows("Rectangle")

elif prediction[0][0] == 'TWO':
    (h, w, d) = image1.shape
    center = (w // 2, h // 2)
```

```
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"):
```



```

elif prediction[0][0] == 'FIVE':
    '''(h, w, d) = image1.shape
    center = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(center, 45, 1.0)
    rotated = cv2.warpAffine(image1, M, (w, h))'''
    gray = cv2.cvtColor(image1, cv2.COLOR_RGB2GRAY)
    cv2.imshow("OpenCV Gray Scale", gray)
    key = cv2.waitKey(3000)
    if (key & 0xFF) == ord("5"):
        cv2.destroyAllWindows()

else:
    continue

interrupt = cv2.waitKey(10)
if interrupt & 0xFF == 27: # esc key
    break

cap.release()
cv2.destroyAllWindows()
return render_template("home.html")

```

## **RUN THE APPLICATION:**

At last, we will run our flask application

```

if __name__ == "__main__":
    # running the app
    app.run(debug=False)

```

Run The app in local browser

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type “python app.py” command
- Navigate to the localhost where you can view your web page

```
C:\Users\Boopathi\anaconda3\envs\ibm_env\python.exe C:/Users/Boopathi/PycharmProjects/pythonProject/IBMProject/Flask/app.py
```

Then it will run on localhost:5000

```
Loaded model from disk
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Navigate to the localhost (<http://127.0.0.1:5000/>) where you can view your web page.

## 8.TESTING :

### 8.1 Test Cases

#### 3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	1	0	0	1
Client Application	1	0	0	1

### 8.2 User Acceptance Testing

## 1. Purpose of Document

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

## 2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	0	0	1
Duplicate	4	1	3	0	8
External	1	3	0	0	4
Fixed	2	4	4	2	12
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	8	8	4	2	22

## 9.RESULTS :

### 9.1 Performance Matrics

#### Locust Test Report

During: 11/17/2022, 4:18:19 PM - 11/17/2022, 4:21:00 PM

Target Host: http://127.0.0.1:5000/

Script: locustfile.py

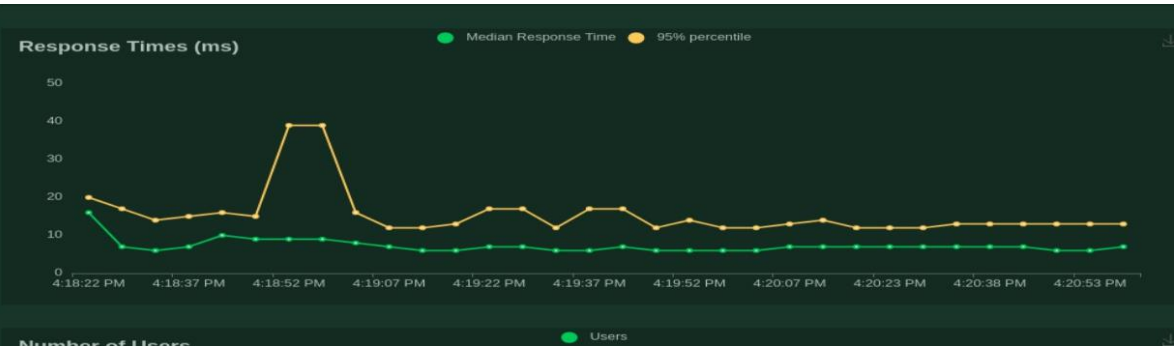
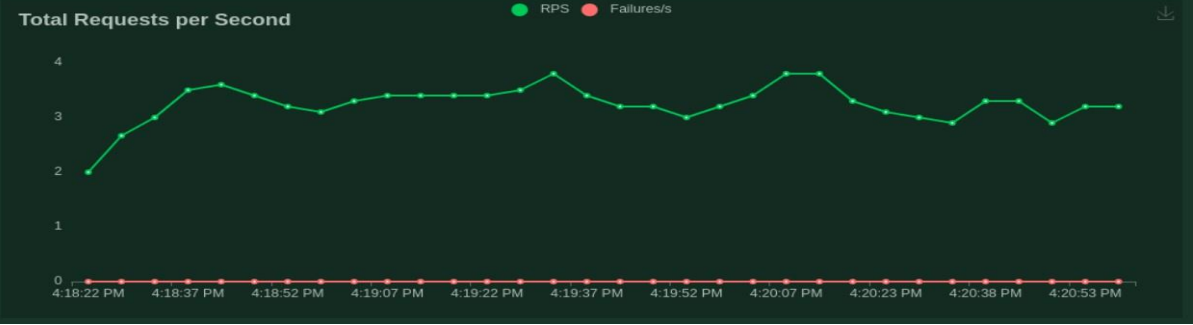
#### Request Statistics

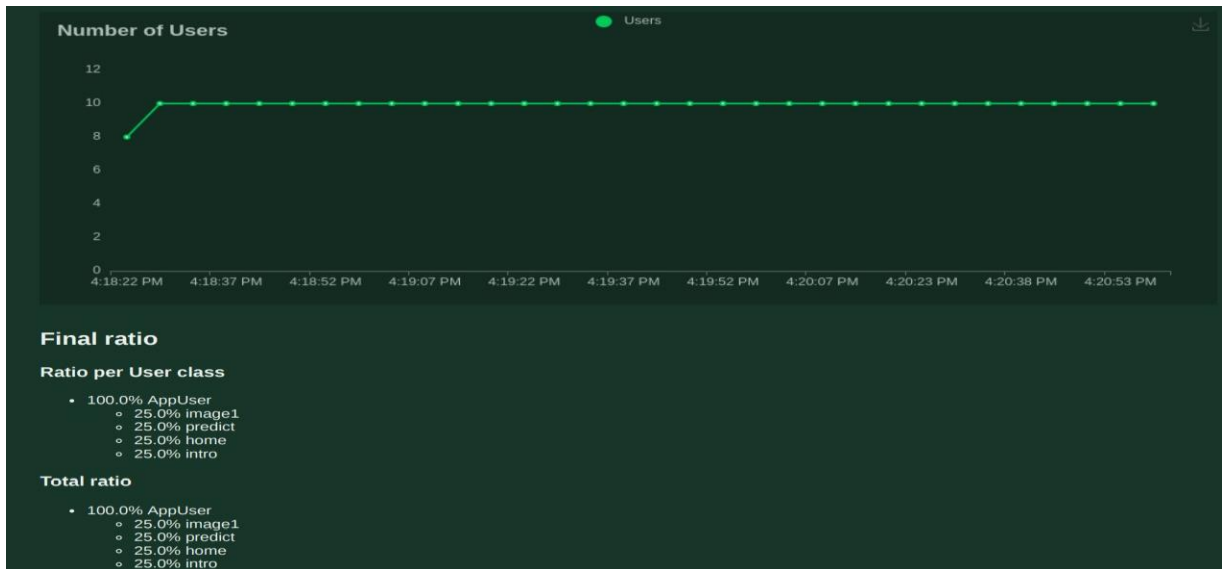
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	126	0	13	9	59	6900	0.8	0.0
GET	/image1	129	0	6	5	27	7394	0.8	0.0
GET	/intro	139	0	7	4	38	8349	0.9	0.0
GET	/predict	136	0	6	5	12	6900	0.8	0.0
Aggregated		530	0	8	4	59	7400	3.3	0.0

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	12	12	13	14	16	17	40	60
GET	/image1	6	6	7	7	8	9	11	27
GET	/intro	6	6	7	7	9	13	22	39
GET	/predict	6	7	7	8	9	10	11	13
	Aggregated	7	7	9	11	12	15	22	60

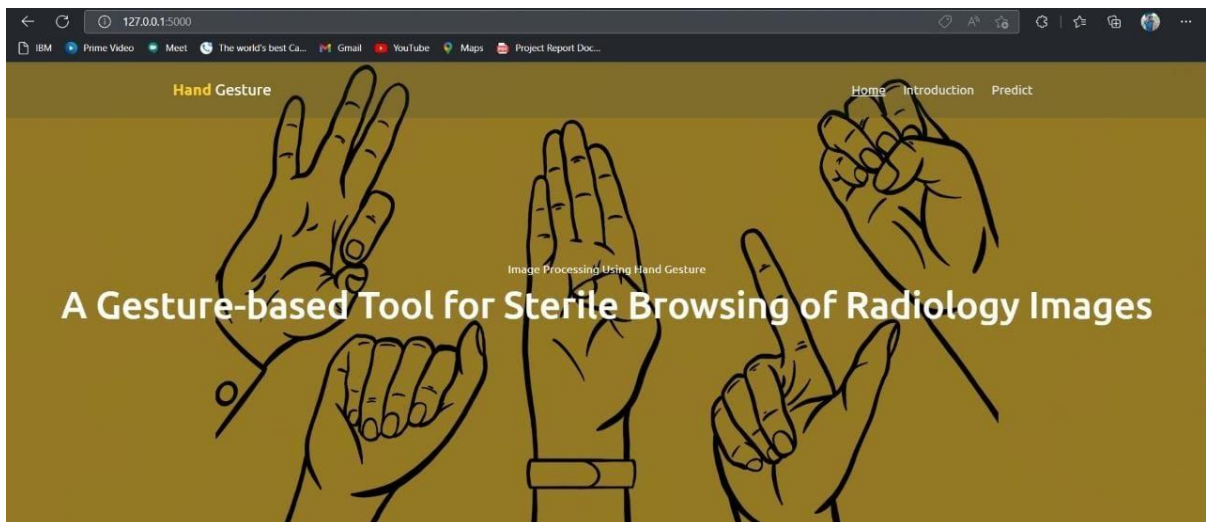
Charts



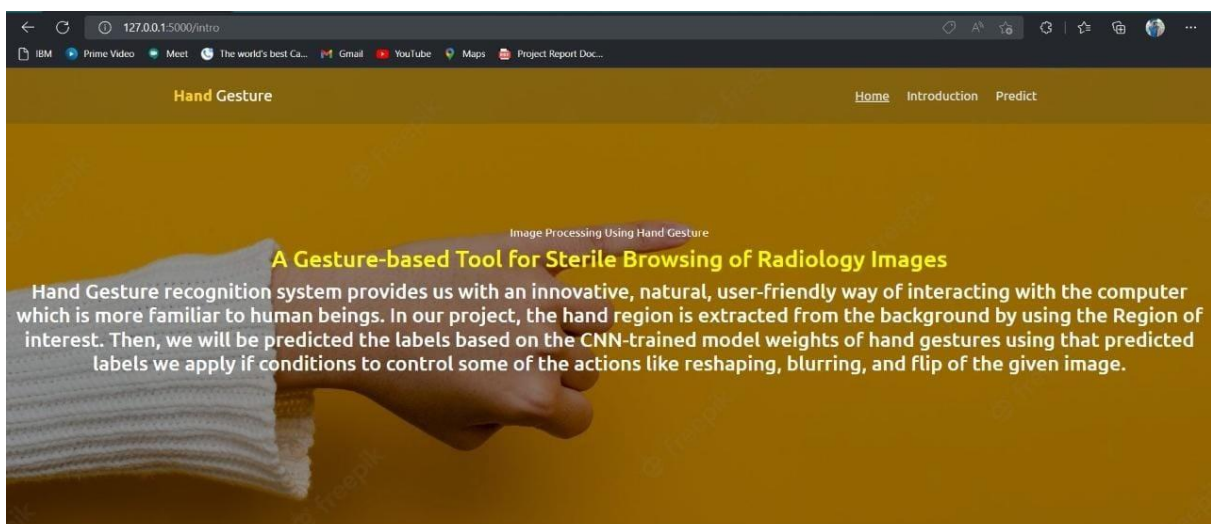


## 9.2 Output

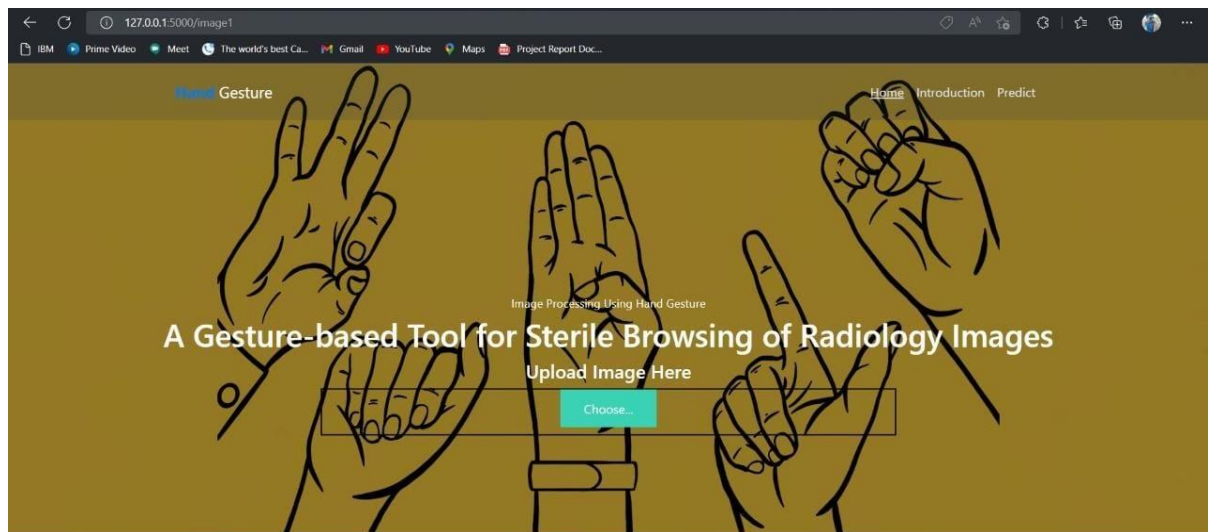
### 1:Home Page:



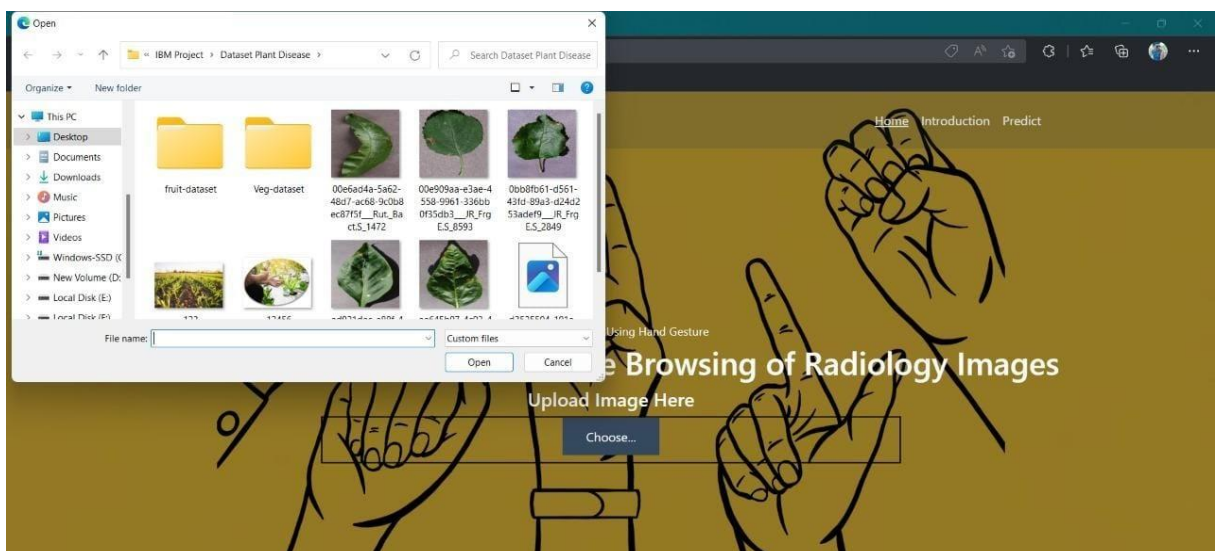
### 2:Introduction Page:



### 3:Predict Page:

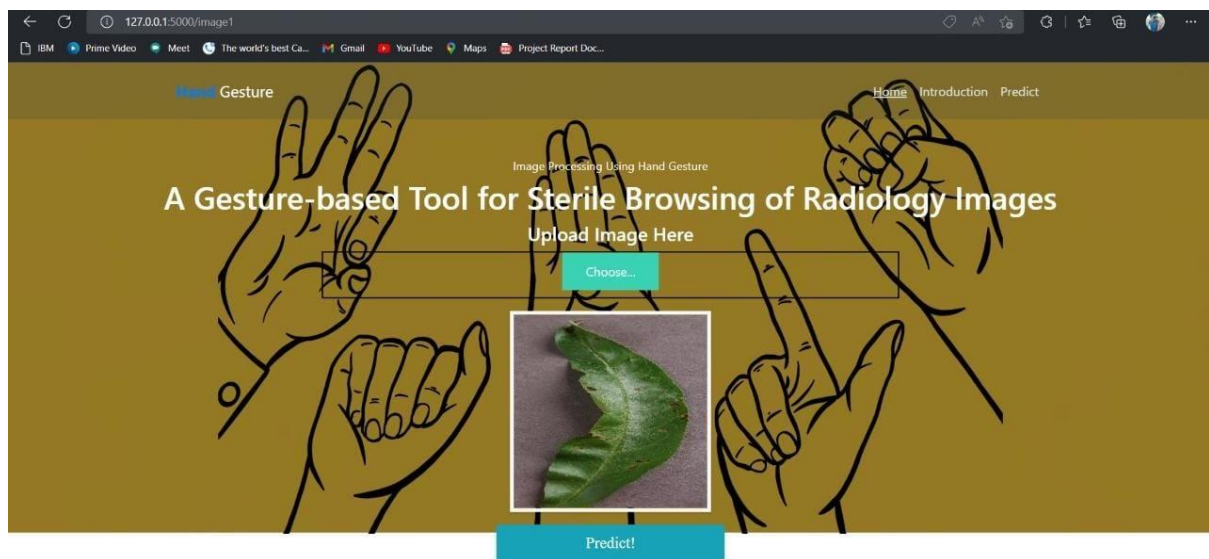


#### 3.1 :Upload Image:

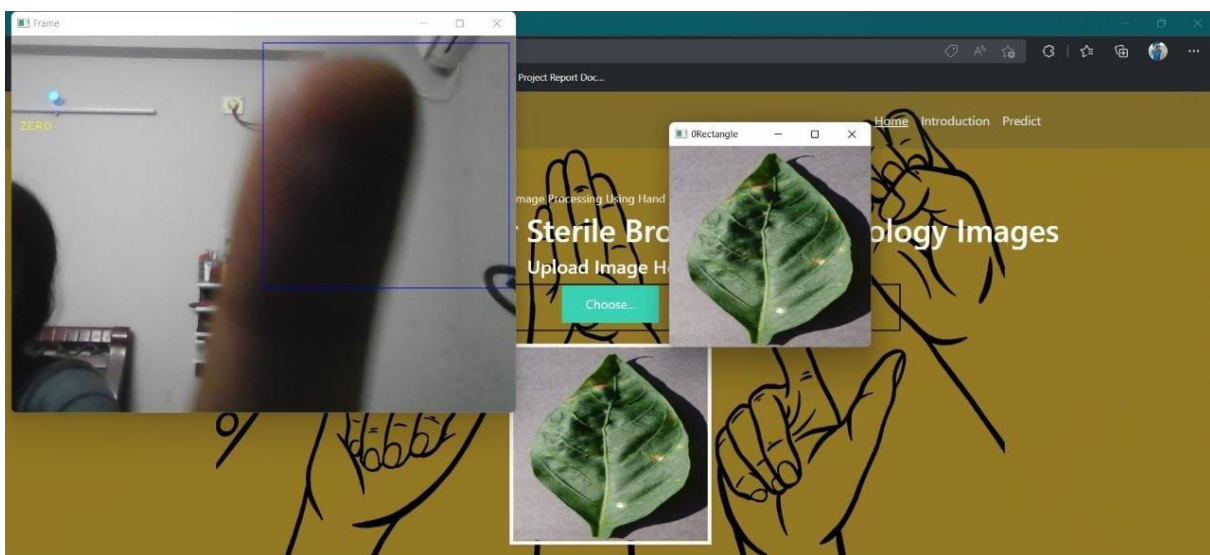


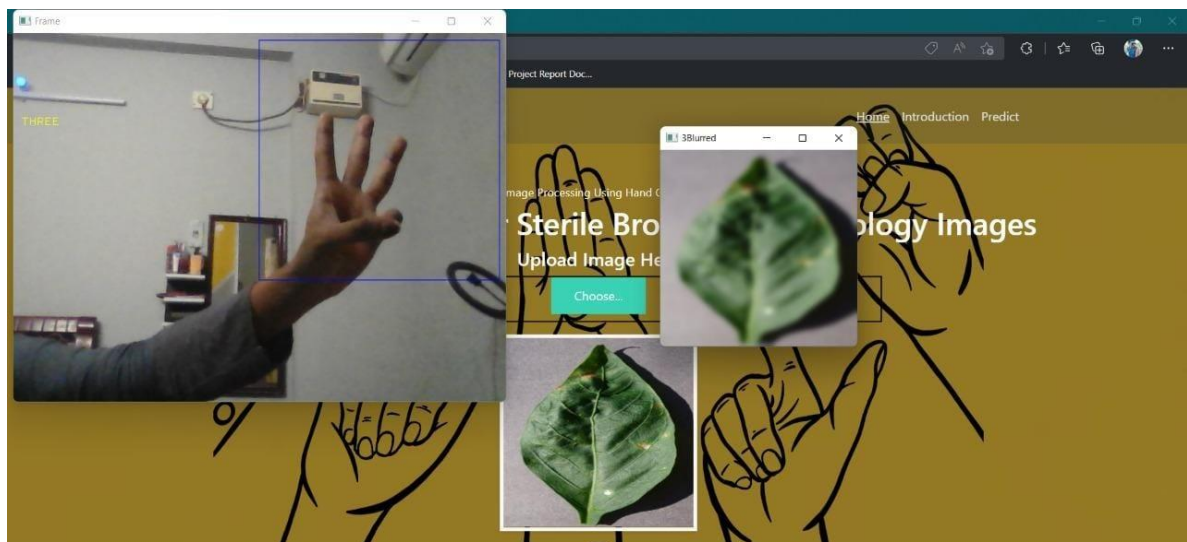
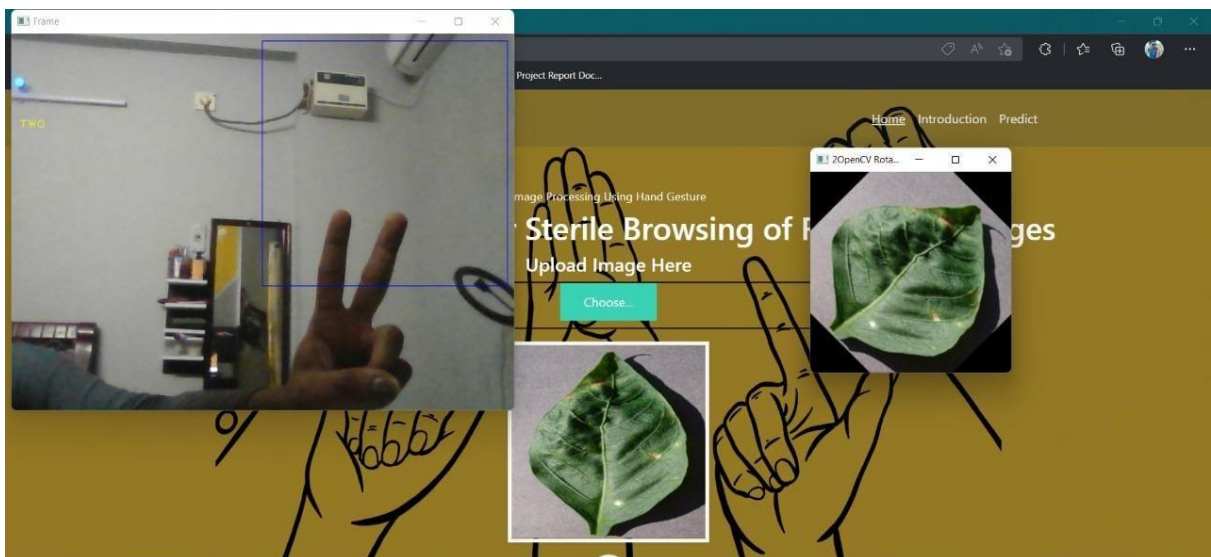
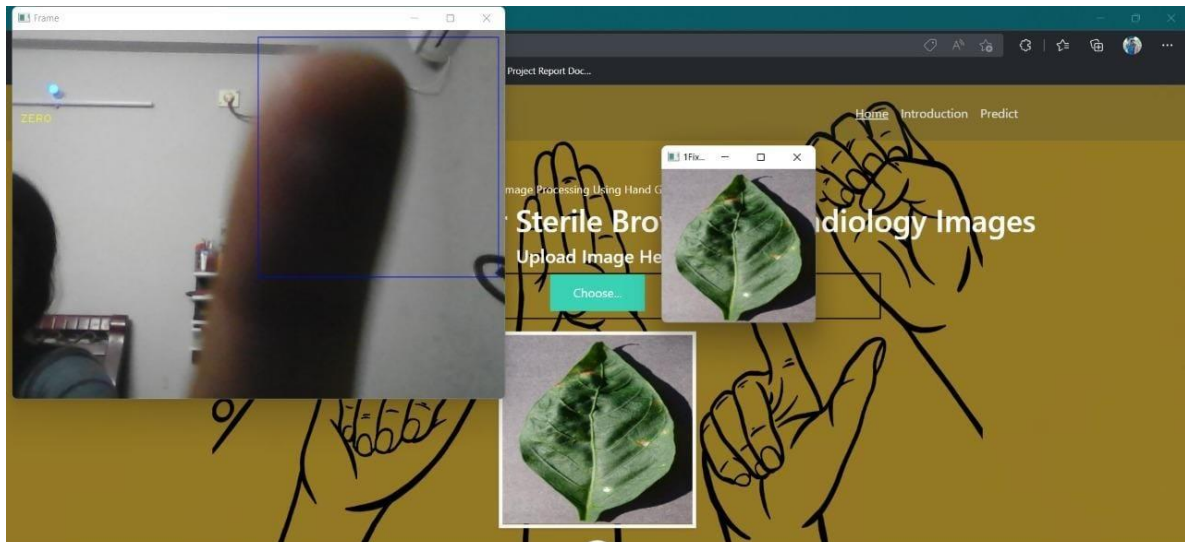


### 3.2:Image Uploaded:

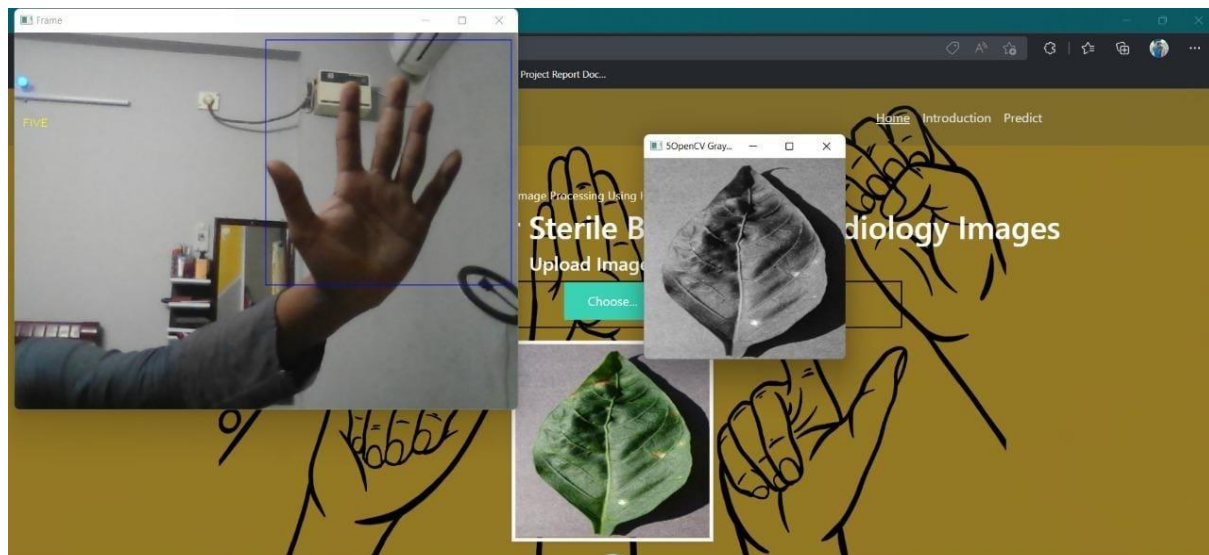
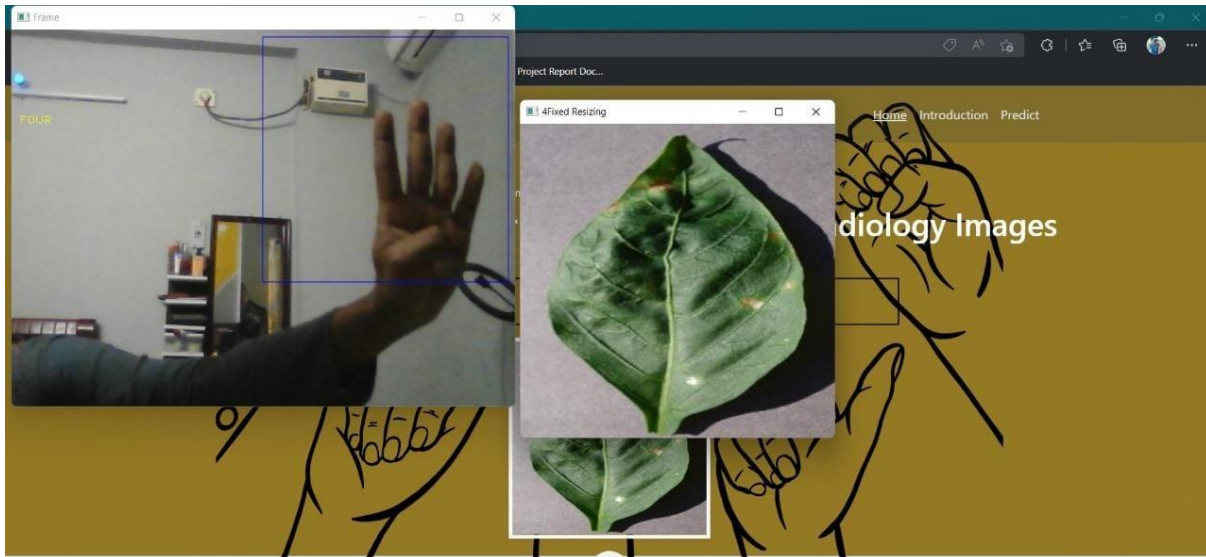


### 3.3 :Predict:



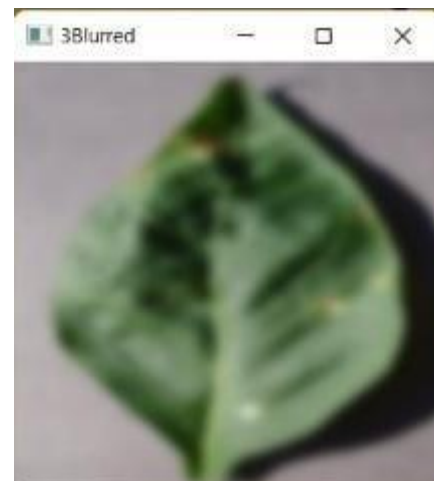






## 4.Actions:

- 0-Rectangle
- 1-Fixed Resizing(200,200)
- 2-OpenCV Rotation
- 3-Blurred
- 4-Fixed Resizing(400,400)
- 5-OpenCV Grey Scale





## 10.ADVANTAGES & DISADVANTAGES:

### 10.1 Advantages:

- **Ease of use**—the system allows the surgeon to use his/her hands, their natural work tool
- **Rapid reaction**—nonverbal instructions by hand gesture commands are intuitive and fast (In practice, the “Gestix” system can process images and track hands at a frame-rate of 150 Hz, thus, responding to the surgeon's gesture commands in real-time)
- **An unencumbered interface**—the proposed system does not require the surgeon to attach a microphone, use head-mounted (body-contact) sensing devices or to use foot pedals
- **Distance control**—the hand gestures can be performed up to 5 meters from the camera and still be recognized accurately.
- Accuracy is high.
- Memory Saving. [ due to cloud ]
- An Unencumbered Interface.
- Distance Control.

Translation independent.

### 10.2 Disadvantages:

- ❖ This kind of input also raises issues that are not relevant with traditional input. On the user's side, these problems are to learn, to remember and to accurately execute gestures.
- ❖ The developer has to provide a system that correctly recognizes these gestures. Freeman et al. remarked that the observation of gestures does not suffice in order to learn them, as the observer is unable to differentiate relevant and irrelevant movements.
- ❖ The teaching of multi-touch and mid-air gestures is more difficult than that of single-touch gestures. In the case of the latter, the hand posture is irrelevant - users only need to follow a path correctly to perform a command.
- ❖ Image Framing is not accurate.
- ❖ Small number of dataset.
- ❖ Wrong prediction in low light.
- ❖ Accuracy is less.
- ❖ Dataset is not suitable for real time.

## **11.CONCLUSION :**

- In this project , we proposed an idea for feasible communication between hearing impaired and normal person with the help of deep learning and machine learning approach. This is ever the surrounding challenge to develop a sign language system in data the collection remains invariant of the unconstraint environment. This project can be extended to the real time data. Our method shows to have potential in tackling this problem employing a straight forward camera as a premade dataset has been used.as been used, in case sufficient substantial training information is given, which can be continuously done and included through the previously mentioned process.

## **12.FUTURE SCOPE :**

- 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.

In future everything become automated .

## 13.APPENDIX :

### 13.1 Source Code

#### 1:home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="Start your development with Creative Design
landing page.">
    <meta name="author" content="Devcrud">
    <title>Hand Gesture Recognition</title>
    <link rel="stylesheet" href="../static/vendors/themify-icons/css/themify-icons.css">
    <link rel="stylesheet" href="../static/css/creative-design.css">
<style>
.header {
    background: #efefef url(../static/imgs/3.jpg);
    background-size: cover;
    background-position: center center;
    background-repeat: no-repeat;
    text-align: center;
    color: white;
    position: relative;
    height: 598px;
    position: relative;
}
</style>
</head>
<body data-spy="scroll" data-target=".navbar" data-offset="40" id="home">
    <!-- Page Navbar -->
    <nav id="scrollspy" class="navbar page-navbar navbar-light navbar-expand-md
fixed-top" data-spy="affix" data-offset-top="20">
        <div class="container">
            <a class="navbar-brand" href="#"><strong
class="text-primary">Hand</strong> <span class="text-dark">Gesture</span></a>
            <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
```

```

        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav ml-auto">
            <li class="nav-item">
                <a class="nav-link" href="{{ url_for('home')}}"><u>Home</u></a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="{{ url_for('intro')}}">Introduction</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="{{ url_for('index6')}}">Predict</a>
            </li>
        </ul>
    </div>
</div>
</nav><!-- End of Page Navbar -->

<!-- Page Header -->
<header id="home" class="header">


    <div class="overlay"></div>
    <div class="header-content">
        <p>Image Processing Using Hand Gesture</p>
        <h1>A Gesture-based Tool for Sterile Browsing of Radiology Images</h1>
    </div>
</header><!-- End of Page Header -->

```

## **2:intro.html**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="Start your development with
Creative Design landing page.">

```

```

    <meta name="author" content="Devcrud">
    <title>Hand Gesture Recognition</title>
    <link rel="stylesheet"
href="../static/vendors/themify-icons/css/themify-icons.css">
    <link rel="stylesheet" href="../static/css/creative-design.css">
<style>
.header {
background: #efefef url(../static/imgs/1.jpg);
background-size: cover;
background-position: center center;
background-repeat: no-repeat;
text-align: center;
color: white;
position: relative;
height: 598px;
position: relative;
}
</style>
</head>
<body data-spy="scroll" data-target=".navbar" data-offset="40"
id="home">
    <!-- Page Navbar -->
    <nav id="scrollspy" class="navbar page-navbar navbar-light
navbar-expand-md fixed-top" data-spy="affix" data-offset-top="20">
        <div class="container">
            <a class="navbar-brand" href="#"><strong
class="text-primary">Hand</strong> <span
class="text-dark">Gesture</span></a>

            <div class="collapse navbar-collapse"
id="navbarSupportedContent">
                <ul class="navbar-nav ml-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="{{
url_for('home')}}"><u>Home</u></a>
                    </li>

```



```

        <li class="nav-item">
            <a class="nav-link" href="{{
url_for('intro')}}">Introduction</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{{
url_for('index6')}}">Predict</a>
        </li>
    </ul>
</div>
</div>
</nav><!-- End of Page Navbar -->

<!-- Page Header -->
<header id="home" class="header">

    <div class="overlay"></div>
    <div class="header-content">
        <p>Image Processing Using Hand Gesture</p>
        <h3 style="color:yellow;">A Gesture-based Tool for Sterile
Browsing of Radiology Images</h3>
        <h4>    Hand Gesture recognition system
provides us with an innovative, natural, user-friendly way of interacting
with the computer which
is more familiar to human beings. In our project, the hand region is
extracted
from the background by using the Region of interest. Then, we will be
predicted
the labels based on the CNN-trained model weights of hand gestures
using that predicted labels we apply if conditions to control some of the
actions like reshaping, blurring, and flip of the given image. <h4>
    </div>
</header><!-- End of Page Header -->

```

**3:index6.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,
shrink-to-fit=no">
  <meta name="description" content="Start your development with
Creative Design landing page.">
  <meta name="author" content="Devcrud">
  <title>Hand Gesture Recognition</title>
  <link rel="stylesheet"
href="../static/vendors/themify-icons/css/themify-icons.css">
  <link rel="stylesheet" href="../static/css/creative-design.css">
  <link href="../static/css/creative-design.css/main.css"
rel="stylesheet">
  <link href="{{ url_for('static', filename='css/main.css') }}"
rel="stylesheet">
  <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"></s
cript>
  <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"></scri
pt>
  <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"></scrip
t>
  <link href="{{ url_for('static', filename='css/main.css') }}"
rel="stylesheet">

```

```

<style>
.header {
  background: #efefef url(../static/imgs/3.jpg);
  background-size: cover;
  background-position: center center;
  background-repeat: no-repeat;
  text-align: center;
  color: white;
  position: relative;
  height: 598px;
  position: relative;
}
.btncls{
  position: relative;
  top: -133px;
  left: 33px;
  border-radius: 20px;
  height:30px;
  font-size: 12px;
  border: none;
  background-color: rgb(45, 45, 43);
  color:rgb(252, 248, 248);
  font-weight: 600;
}
#display-image{
  width: 400px;
  height: 225px;
  border: 3px solid rgb(45, 45, 43);
  border-radius:25px;
  background-position: center;
  background-size: cover;
  position: relative;
  top: -143px;
  left: 33px;
}
#image-input{

```

```
position: relative;
top: -153px;
left: 33px;
border-radius: 20px;
background-color: rgb(45, 45, 43);
}

.topnav {
  overflow: hidden;
  background-color: #056959;
}

.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;
}
```

```
.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);
}
form {border: 2px solid black; margin-left:400px;margin-right:400px;}

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;
}

button {
    background-color: #091425;
    color: black;
    padding: 14px 20px;
    margin-bottom:10px;
    border: none;
    cursor: pointer;
    width: 19%;
    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;

}
.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;
}

.active {
    background-color: #267481;
}

/* 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 data-spy="scroll" data-target=".navbar" data-offset="40"
id="home">
  <!-- Page Navbar -->
  <nav id="scrollspy" class="navbar page-navbar navbar-light
navbar-expand-md fixed-top" data-spy="affix" data-offset-top="20">
    <div class="container">
      <a class="navbar-brand" href="#"><strong
class="text-primary">Hand</strong> <span
class="text-dark">Gesture</span></a>
      <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse"
id="navbarSupportedContent">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">

```

```

                <a class="nav-link" href="{{
url_for('home')}}"><u>Home</u></a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="{{
url_for('intro')}}">Introduction</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="{{
url_for('index6')}}">Predict</a>
            </li>
        </ul>
    </div>
</div>
</nav><!-- End of Page Navbar -->

<!-- Page Header -->
<header id="home" class="header">


    <div class="overlay"></div>
    <div class="header-content"><br><br><br><br><br><br><br>
        <p>Image Processing Using Hand Gesture</p>
        <h1>A Gesture-based Tool for Sterile Browsing of Radiology
Images</h1>
        <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" color="#f35ac5" 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>

```

## 13.2 GitHub & Project Demo Link

### PROJECT GITHUB:

<https://github.com/IBM-EPBL/IBM-Project-26433-1660026553>

### PROJECT DEMO LINK:

<https://drive.google.com/file/d/19Rhnb1FLsmfVVIDHEZ7laHzt7bcdE0X6/view?usp=drivesdk>