A GESTURE-BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

Bonafide record of work done by

PAVANSAI.V - 811519104082

VITHUN.V.C - 811519104121

MOHANRAJ.S - 811519104072

MOHAMMED FAIS.M - 811519104071

TEAM ID : PNT2022TMID10905

INDUSTRY MENTOR : PRADEEPTHI

FACULTY MENTOR : SWATHI

BACHELOR OF ENGINEERING

BRANCH: COMPUTER SCIENCE AND ENGINEERING

NOVEMBER 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS INSTITUTION)

SAMAYAPURAM, TRICHY – 621 112

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

In this project, we will use gestures to browse images acquired in the radiology department. Gestures refer to non-verbal communication with hands. A major challenge in this process is to provide physicians with an efficient, intuitive, accurate and safe way to interact without compromising the quality of their work.

But in critical care units (ICUs), using computer keyboard and mouse is a typical way for medical professionals to transfer illnesses. In this article, we propose using hand gestures as an alternative to current interface techniques, with the key benefit of sterility. Humans are adept at deciphering both body and sign language. This may be a result of how the brain developed in conjunction with synaptic contacts and eyesight.

There are a few issues that need to be resolved in order to reproduce this ability in computers, including how to distinguish things of interest in photographs and which image capture method and classification technique are more suitable.

In this project A Gesture-based Tool for Sterile Browsing of Radiology Images, First the model is trained pre trained on the images of different hand gestures, such as a showing number with fingers as 1 ,2,3,4. This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with the pre-trained model and the gesture is identified. If the gesture predicts is 1; then images are blurred; 2, image is resized; 3, image is rotated etc.

1.2 PURPOSE

1. Without using the keyboard or mouse physically, the application can be used to interact with the application remotely.

2. The gesture tool can understand the natural way that a person and computer communicate by employing finger movements spread out over a brief period of time.

3. Many doctors can work more efficiently thanks to this gesture-based project.

4. Due to the fact that all commands were carried out using hand gestures, the doctors didn't need to move once during the whole procedure.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. Providing doctors with effective, intuitive, accurate, and safe ways of contact without compromising the caliber of their job is a significant problem.

2. Some problems with the system, such as slow performance, may arise when accessing the camera.

3. The user must be able to distinguish between objects of interest in images; capture technology and classification approach are more suitable among other things.

2.2 REFERENCES

1. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor Zhou Ren, Jun song Yuan, Member, IEEE, Jingjing Meng, Member, IEEE, and Zheng you Zhang, Fellow, IEEE, 15, AUGUST 2013.

2. A Fast Gesture Recognition Scheme for Real-Time Human Machine Interaction Systems . Ching-Hao Lai* Smart Network System Institute for Information Industry Taipei City, Taiwan , 2010.

3. Intension, Context and Gesture Recognition for Sterile MRI Navigation in the Operating Room by Agency for Healthcare Research and Quality (AHRQ)

4. Hand Gestures Recognition Using Radar Sensors for Human-Computer Interaction Supported by the Bio ad Medical Technology Development Program of the National Research Foundation(NRF)

5. 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.

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

2.3 PROBLEM STATEMENT DEFINITION

1. The webcams must be capable of identifying motion gestures that have already been programmed into the system.

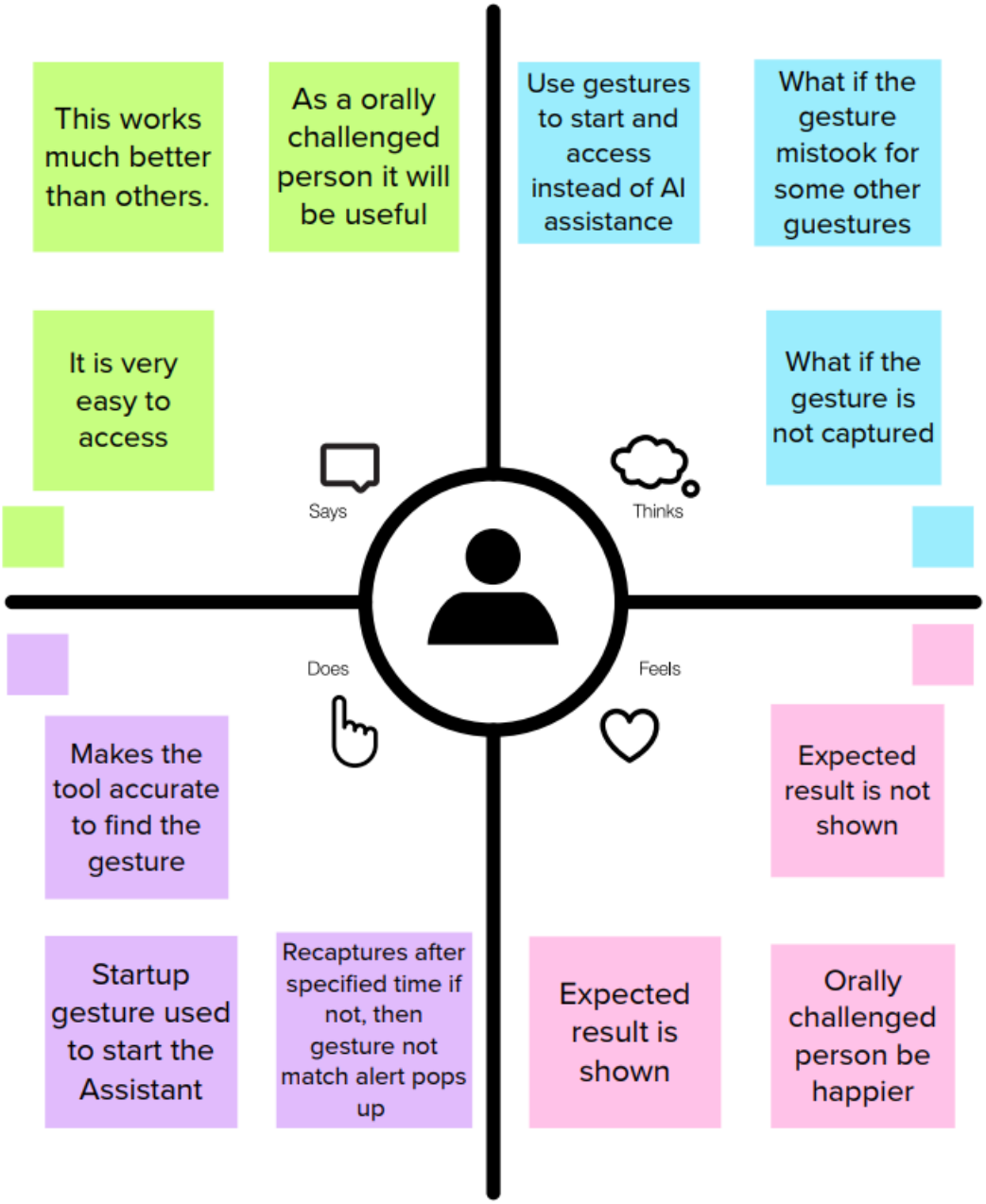
2. The program must be available to the user wherever they are and whenever they need it.

3. The user must be able to upload pictures of different sizes, blur them, resize them, and rotate them. This instrument needs to be able to communicate with people and comprehend symbols.

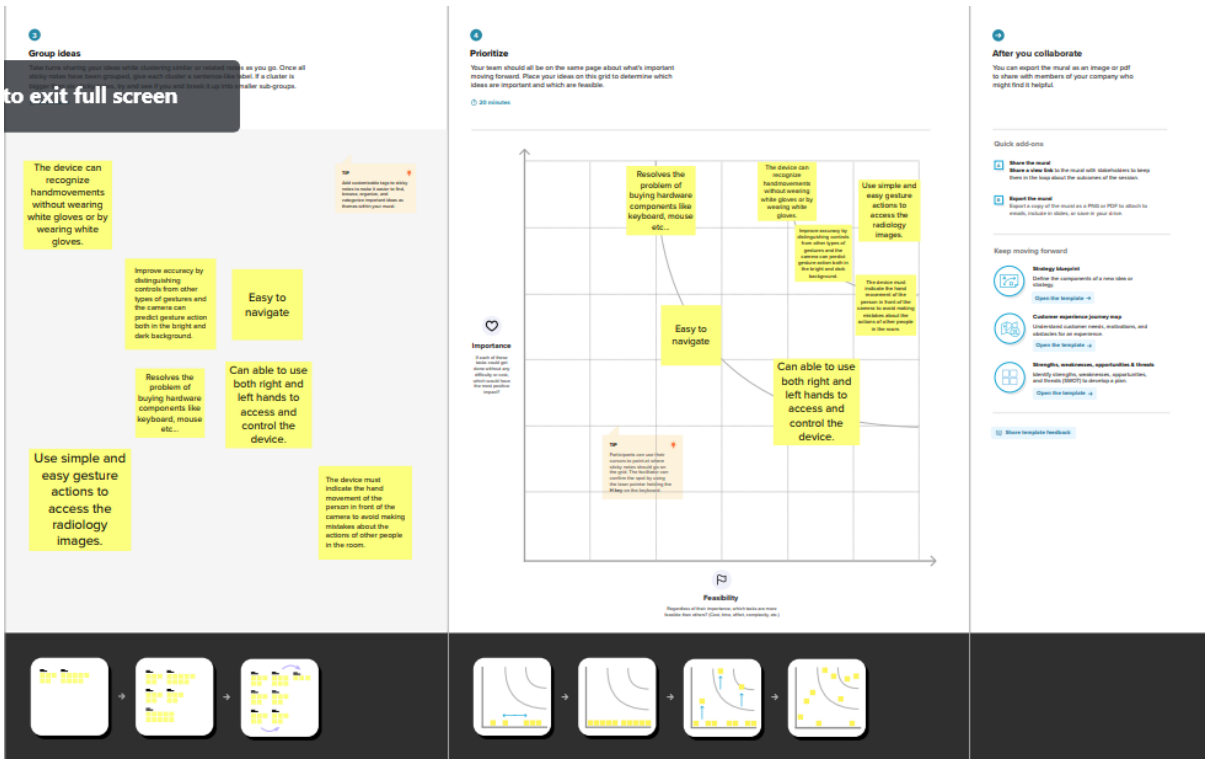
4. The machine must be able to run multiple tasks continuously, without interruption.

CHAPTER 3
IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



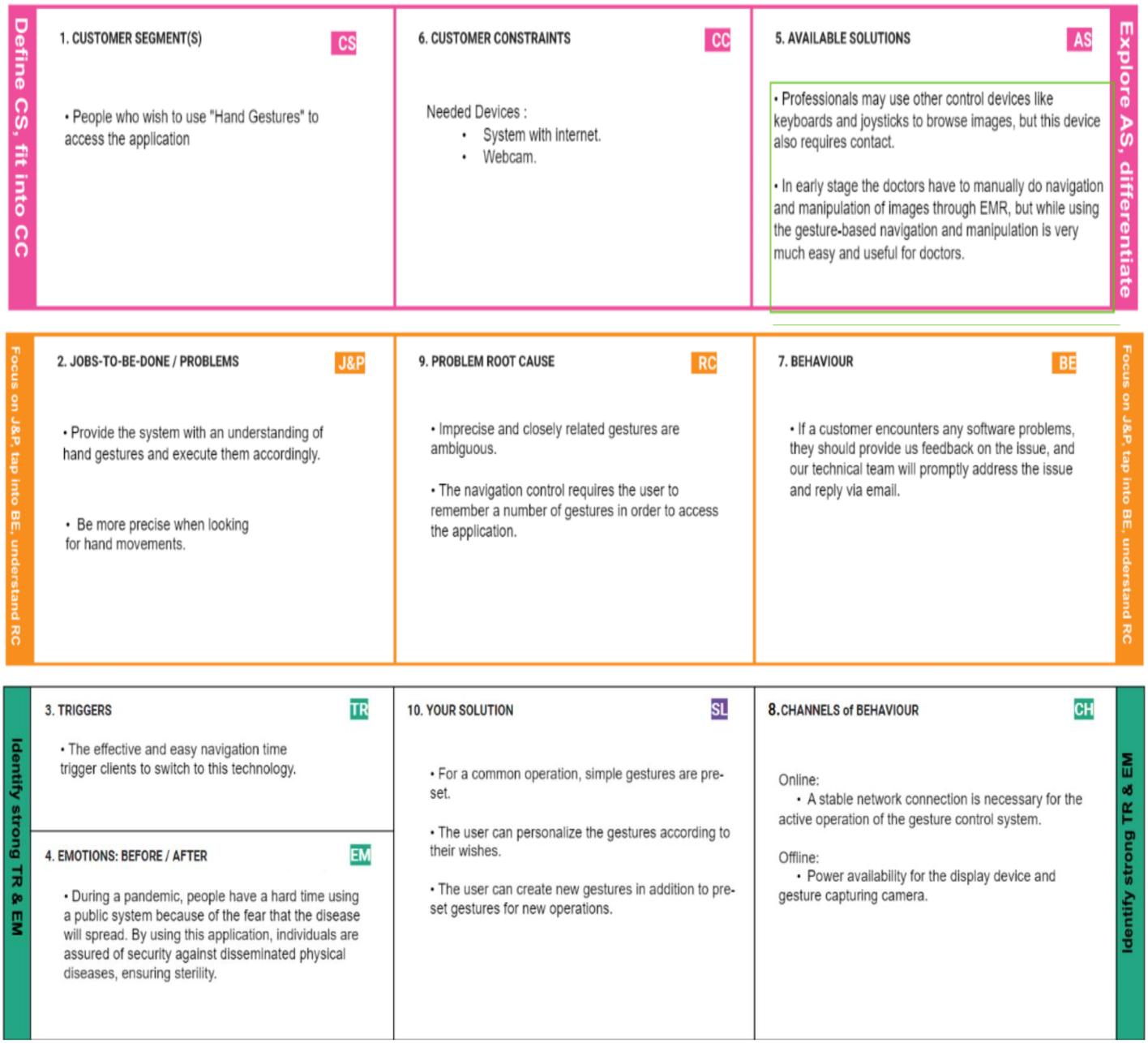
3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• In order a computer to recognize a sign language (gesture) and perform the operation that are predefined for the specific gesture
2.	Idea / Solution description	• For common operation simple gestures are predefined. • The user can customise the gestures according to his wish. • The user can create new gesture apart from the predefined gestures for new operations.
3.	Novelty / Uniqueness	• Resolves the problem of buying hardware components except camera like keyboard, mouse etc... • Camera can predict the gesture action both in bright and dim background. • Require less usage of battery.
4.	Social Impact / Customer Satisfaction	• Orally challenged persons can use gestures to perform operations like normal person uses personal assistant(virtual) • Surgeons in operation theatre can use devices using gestures without physical contact
5.	Business Model (Revenue Model)	• Orally challenged persons find devices with gesture-based tool more useful for them • Kids and youngsters can be attracted by using gesture-based tools. • By using gesture-based tools in public places like shopping mall, theatre can attract more customers.
6.	Scalability of the Solution	• Demand for devices with gesture-based tools for orally challenged persons is more. • Devices with gesture-based tools are highly in demand in places where hygiene plays a major role like hospital, infant care centres etc... • Devices with gesture-based tools will be more attractive among youngsters.

3.4 PROBLEM SOLUTION FIT



CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Interface	The user-friendly interface, which helps to upload customised gesture images for easy Human Computer Interaction.
FR-4	Add image option	Add image through Device
FR-5	Gesture Operation	Dynamic navigation gestures are translated to commands based on their relative positions on the screen.

4.2 NON-FUNCTIONAL REQUIREMENT

Non-functional Requirements:

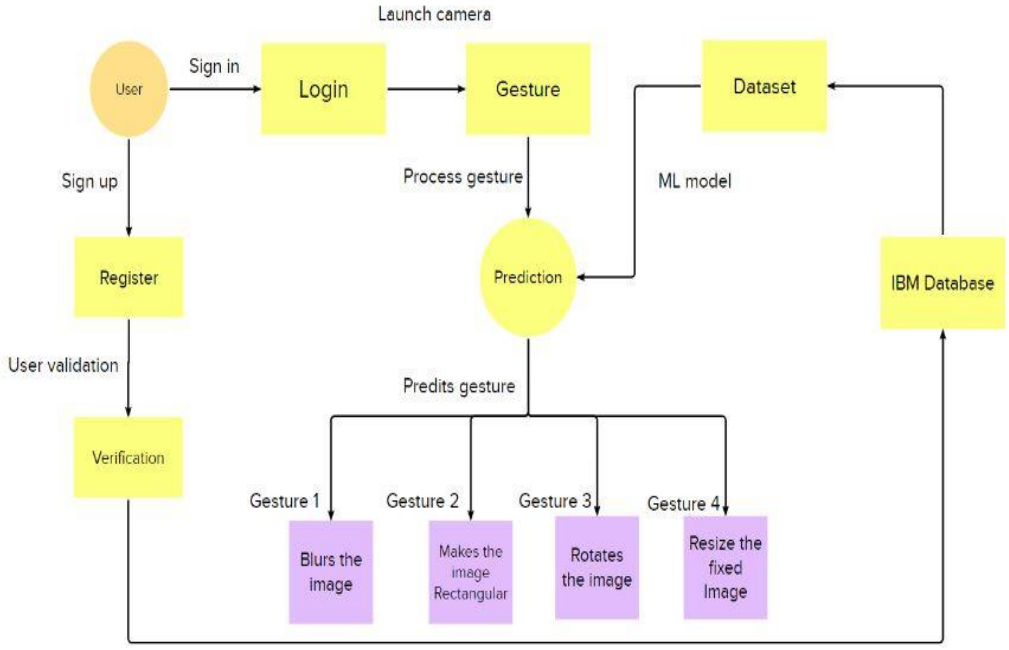
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The Application is easy to access since there is less physical contact with the system.
NFR-2	**Security**	The user will have unique credentials to sign in their account and further they can secure with two factor authentication if needed.
NFR-3	**Reliability**	The gestures are identified accurately by the application in all lighting conditions using ML and AI algorithms.
NFR-4	**Performance**	The application recognizes the gesture in few milliseconds and the corresponding operation is performed.
NFR-5	**Availability**	The new Application, which is free to download, requires no special hardware, just a standard web camera.
NFR-6	**Scalability**	As the Application is cloud based, ample amount of user can use them.

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.



5.2 SOLUTION &TECHNICAL ARCHITECTURE

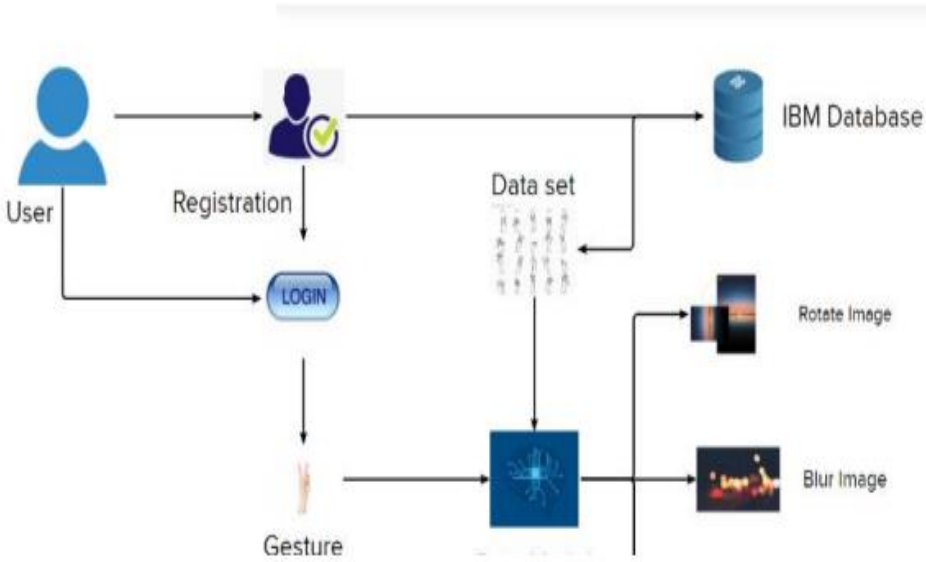
5.2.1 SOLUTION ARCHITECTURE:

1. Different operations are applied to the input image depending on the various gesture inputs.

2. You can crop, blur, flip, and resize the image.

3. After the model has analyzed the gesture, the prediction with the operation done to the image is displayed on the user interface.

4. Better execution in terms of accuracy, sensitivity, system architecture design, and software flexibility.



5.2.2 TECHNOLOGY ARCHITECTURE:

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application Web UI,.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	NA	NA
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	NA	NA
9.	External API-2	NA	NA
10.	Machine Learning Model	Purpose of Machine Learning Model	Gesture Recognition Model,
11.	Infrastructure (Server / Cloud)	Cloud	IBM Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used FLASK	Technology of Opensource framework PYTHON
2.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services) This improves scalability, because application servers can be deployed on many machines. The database does not make longer connections with every client – it only requires connections from a smaller number of application servers	Presentation Layer – FLASK (HTML, CSS) Application Layer – Flask (Python) Data Layer – IBM DB2
3.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	-
4.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	-

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer		USN-1	As a user, I enter the home page, I face a welcoming sentence and there was also an intro button and launch button.		Low	Sprint-1
		USN-2	As a user, I click the intro button, it navigates to the intro page where I get a simple introduction about the application.		Medium	Sprint-2
		USN-3	As a user, I click the launch button, it navigates to the launch page where I can upload a picture.		High	Sprint-3
		USN-4	As a user, after I upload the picture, I click the predict button to for further process.		High	Sprint-3
		USN-5	As a user, after I click predict button, the camera opens and I show the gesture to modify the picture.		High	Sprint-4

CHAPTER 6
PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		USN-1	As a user, I enter the home page, I face a welcoming sentence and there was also an intro button and launch button.	1	Low	PAVANSAI.V, VITHUN.V.C, MOHAN RAJ.S, MOHAMMED FAIS.M
Sprint-2		USN-2	As a user, I click the intro button, it navigates to the intro page where I get a simple introduction about the application.	1	Medium	PAVANSAI.V, VITHUN.V.C, MOHAN RAJ.S, MOHAMMED FAIS.M
Sprint-3	The user uploads a picture.	USN-3	As a user, I click the launch button, it navigates to the launch page where I can upload a picture.	2	High	PAVANSAI.V, VITHUN.V.C, MOHAN RAJ.S, MOHAMMED FAIS.M
Sprint-3		USN-4	As a user, after I upload the picture, I click the predict button to for further process.	2	High	PAVANSAI.V, VITHUN.V.C, MOHAN RAJ.S, MOHAMMED FAIS.M
Sprint-4	The user uses gestures to perform operations.	USN-5	As a user, after I click predict button, the camera opens and I show the gesture to modify the picture.	2	High	PAVANSAI.V, VITHUN.V.C, MOHAN RAJ.S, MOHAMMED FAIS.M

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	06 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	15	20 Nov 2022

6.3 REPORTS FROM JIRA

Velocity:

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

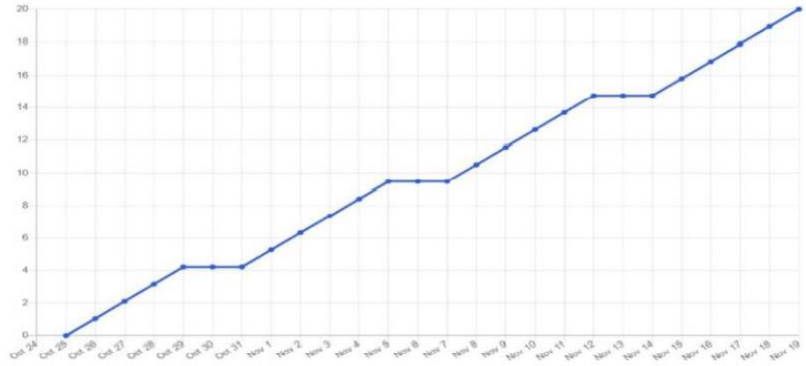
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

$$AV = 18.75/6$$

$$AV = 3.125$$

Burndown Chart:

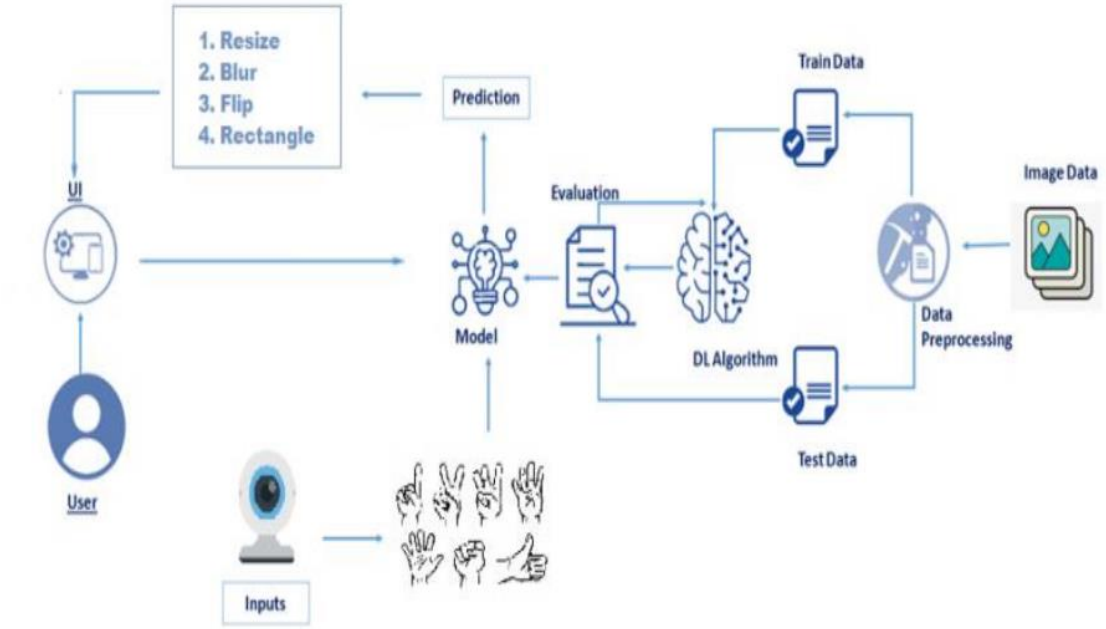
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



CHAPTER 7
CODING & SOLUTIONING

7.1 FEATURE

FR NO	Feature	Description
FR-1	Dataset	Sample Dataset for Training purpose
FR-2	Choose the Image	Can upload the image to be modified
FR-3	Prediction	Predicts the Hand Gesture
FR-4	Image Manipulation	Changes the input images based on the Gesture



CHAPTER 8
TESTING

8.1 TEST CASES

Section	Total Cases
Print Engine	9
Client Application	48
Security	3
Outsource Shipping	4
Exception Reporting	10
Final Report Output	4
Version Control	5

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 [ProductName] 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	13	4	3	3	23
Duplicate	1	0	3	0	4
External	2	2	0	1	5
Fixed	15	4	4	20	4
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	31	29	14	26	47

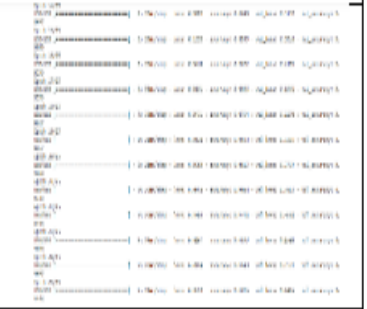
3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	9	0	0	9
Client Application	48	0	0	48
Security	3	0	0	3
Outsource Shipping	4	0	0	4
Exception Reporting	10	0	0	10
Final Report Output	4	0	0	4
Version Control	5	0	0	5

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Detects the hand gesture shown by the user	
2.	Accuracy	Training Accuracy - 98% Validation Accuracy – 93%	
3.	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	NA

CHAPTER 10
ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. This tool's ability to maintain the sterility of the environment is one of its main benefits.

2. Additionally, compared to the current methods of browsing images, it is quicker and easier to use.

3. Time can be saved by performing it even if the surgeon is a little distance from the system.

4. The individual using the tool does not need to have any equipment or other gadgets with them in order to use it. To browse the photographs, they only need to move their hands.

DISADVANTAGES:

1. The tool can be quite pricey because it needs expensive cameras and other equipment to capture and process images.

2. Such systems are challenging to create due to their complexity and high implementation costs.

3. This system is not platform independent because each gesture has a unique control command associated with it, and some control commands change depending on the operating system.

CHAPTER 11
CONCLUSION

In this project, we created a program that can recognize hand movements and let doctors use them to view radiography images. As they won't have to use a mouse or keyboard to browse the images, doctors and surgeons can keep their workspace sterile. In addition to being simple to use, this tool is quicker than the standard mouse/keyboard technique. Since the user doesn't need to be in contact with any equipment, it can be utilized wherever they are. Additionally, the user does not need to bring any equipment with them in order to use it. This technology can also be applied to other fields, such as those where presenters and teachers need to exhibit graphics in the classroom.

CHAPTER 12
FUTURE SCOPE

1. The tool can be made quicker by increasing the recognition speed.

2. The functionality and suitability of this tool for various uses can be increased by adding more gestures.

3. The number of commands can be expanded by adding tracking for both hands.

4. To further enhance the functionality, voice commands can be added.

APPENDIX

A.1 SOURCE CODE

PYTHON CODE:

```python
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
import numpy as np

from tensorflow.python.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')
print("Loaded model from disk")

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

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


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

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


@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def predict():
    if request.method == 'POST':
        print("inside image")
        f = request.files['image']
```

```python
@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def predict():
    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)
        cap = cv2.VideoCapture(0)
        while True:
            _, frame = cap.read() #capturing the video frame values
            # Simulating mirror image
            frame = cv2.flip(frame, 1)

            # Got this from collect-data.py
            # 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],
```

```
        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
```

```
        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':
            '''(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.destroyWindow("OpenCV Gray Scale")
```

```
                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("home.html")

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

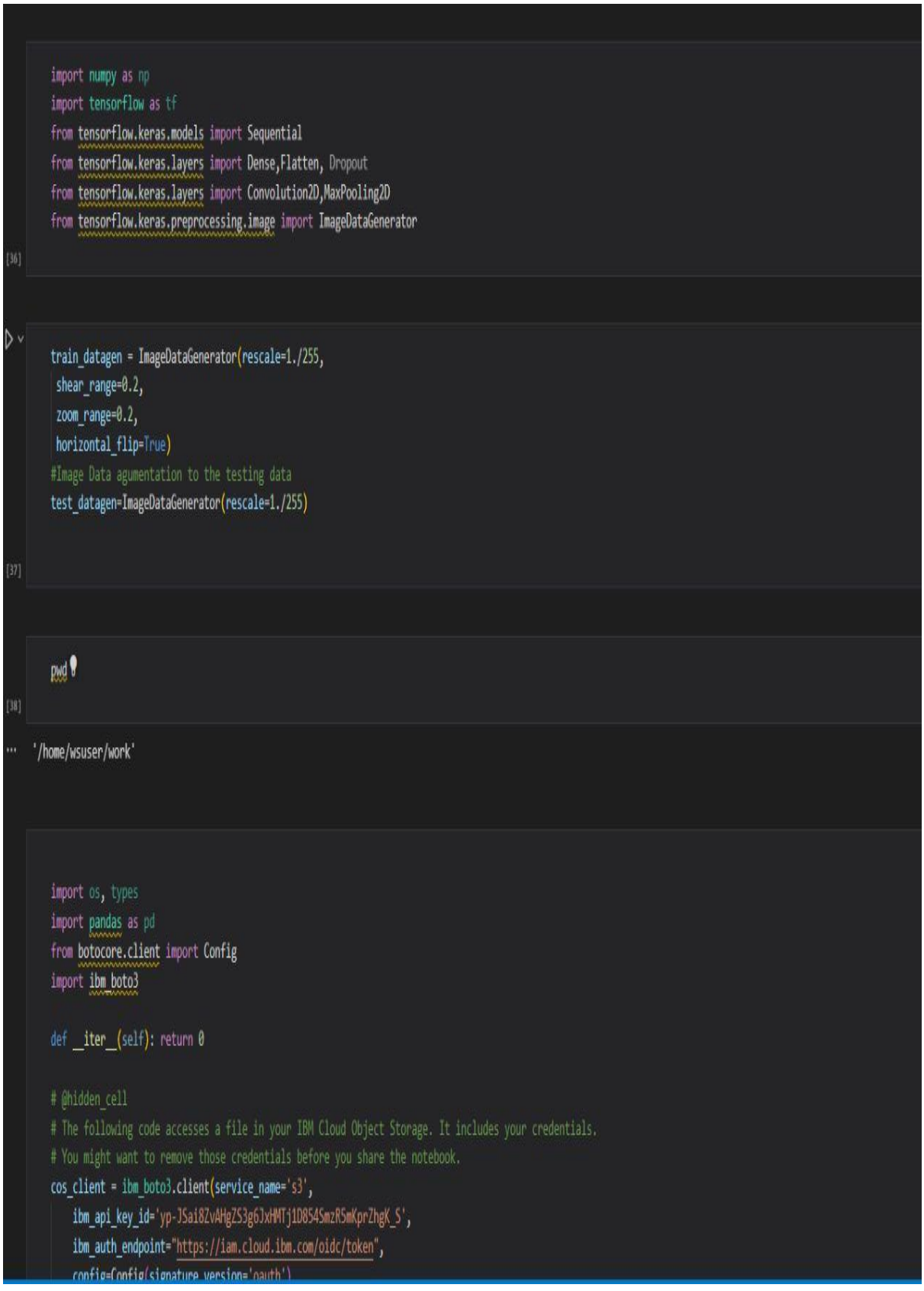
HTML CODE:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static' , filename='CSS/home.css') }}">
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@700&family=Poor+Story&display=swap" rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@700&family=Pacifico&family=Poor+Story&display=swap" rel="stylesheet">
    <title>Home</title>


</head>
<body>
<div class="topnav">
  <span>Hand Gesture System</span>
  <a href="/image1">Launch</a>
  <a href="/intro">Intro</a>
  <a class="active" href="/">Home</a>
</div >

<div class="box">
    <div class="gesture">
        <p>A <br><br>Gesture BasedTool <br><br>for Sterile Browsing Of <br><br>Radiology Images</p>
    </div>

    <!-- <div class="frame">
        <iframe width="420" height="220" src="https://www.youtube.com/embed/nD621G8u6oc?start=3&loop=1&autoplay=1&mute=1&controls=0">
        </iframe>
        <p class="demo">Demonstration Video</p>
    </div> -->
</div>
```

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static' , filename='CSS/intro.css') }}">
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@700&family=Poor+Story&display=swap" rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@700&family=Pacifico&family=Poor+Story&display=swap" rel="stylesheet">
    <title>Intro</title>
</head>

<body>
<div class="topnav">
  <span>Hand Gesture System</span>
  <a href="/image1">Launch</a>
  <a class="active" href="/intro">Intro</a>
  <a  href="/">Home</a>
</div>

<div class="title">
    <p class="intro">INTRODUCTION</p>
</div>

<div class="container">
<span class="content">
      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.

</span>
</div>


<center>
```

```
</style>
</head>

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

   <a href="home.html">Home</a>
   <a href="/intro">Introduction</a>
   <a class="active" href="/image1"><u>Launch</u></a>
 </div>
</div>
<br>

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

       <footer>
   <!-- <script src="main.js" type="text/javascript"></script>     -->
</footer>
```

ipynb CODE:



```
import numpy as np
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
```

[36]

```
train_datagen = ImageDataGenerator(rescale=1./255,
 shear_range=0.2,
 zoom_range=0.2,
 horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

[37]

```
pwd
```

[38]

```
'/home/wsuser/work'
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='yp-JSai8ZvAHgZS3g6JxHMTj1D854SmzR5mKprZhgK_5',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth')
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='yp-JSai8ZvAHgZS3g6JxHMTj1D854SmzR5mKprZhgK_S',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'agesturebasedtoolforsterilebrowsi-donotdelete-pr-rpcywafuepgm9v'
object_key = 'drive-download-20221029T044045Z-001.zip'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

```
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

```
import numpy as np
#open source used for both ML and DL for computation
import tensorflow as tf
```

```
import numpy as np
#open source used for both ML and DL for computation
import tensorflow as tf
#it is a plain stack of layers
from tensorflow.keras.models import Sequential
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten, Dropout
#Falten-used fot flattening the input or change the dimension,
from tensorflow.keras.layers import Convolution2D,MaxPooling2D
#Its used for different augmentation of the image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
train_datagen = ImageDataGenerator(rescale=1./255,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
x_train =train_datagen.flow_from_directory(r'/home/wsuser/work/drive-download-20221029T044045Z-001/train',
target_size=(64, 64),
batch_size=3,
color_mode='grayscale',
class_mode='categorical')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(r'/home/wsuser/work/drive-download-20221029T044045Z-001/test',
target_size=(64, 64),
batch_size=3,
color_mode='grayscale',
class_mode='categorical')
```

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

```
# Initializing the CNN
model = Sequential()
# First convolution layer and pooling
model.add(Convolution2D(32, (3, 3), input_shape=(64, 64, 1),
activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
# Second convolution layer and pooling
model.add(Convolution2D(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 i.e. input layer
model.add(Flatten())
# Adding a fully connected layer, i.e. Hidden Layer
model.add(Dense(units=512 , activation='relu'))
# softmax for categorical analysis, Output Layer
model.add(Dense(units=6, activation='softmax'))
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 62, 62, 32)        320

max_pooling2d_2 (MaxPooling  (None, 31, 31, 32)        0
2D)

conv2d_3 (Conv2D)            (None, 29, 29, 32)        9248

max_pooling2d_3 (MaxPooling  (None, 14, 14, 32)        0
2D)

flatten_1 (Flatten)          (None, 6272)              0

dense_2 (Dense)              (None, 512)               3211776

dense_3 (Dense)              (None, 6)                 3078
```

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

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

```
model.fit_generator(x_train,
steps_per_epoch = 594/3 ,
epochs = 25,
validation_data = x_test,
validation_steps = 30/3 )
```

```
Epoch 1/25

/tmp/wsuser/ipykernel_209/238739696.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,

Output exceeds the size limit. Open the full output data in a text editor
198/198 [==============================] - 13s 65ms/step - loss: 1.2816 - accuracy: 0.5219 - val_loss: 0.5961 - val_accuracy: 0.7667
Epoch 2/25
198/198 [==============================] - 13s 63ms/step - loss: 0.6114 - accuracy: 0.7660 - val_loss: 0.4151 - val_accuracy: 0.8667
Epoch 3/25
198/198 [==============================] - 13s 64ms/step - loss: 0.4101 - accuracy: 0.8215 - val_loss: 0.3007 - val_accuracy: 0.9333
Epoch 4/25
198/198 [==============================] - 12s 63ms/step - loss: 0.2901 - accuracy: 0.8906 - val_loss: 0.2353 - val_accuracy: 0.8333
Epoch 5/25
198/198 [==============================] - 13s 64ms/step - loss: 0.2714 - accuracy: 0.9057 - val_loss: 0.2561 - val_accuracy: 0.9000
Epoch 6/25
198/198 [==============================] - 12s 62ms/step - loss: 0.1690 - accuracy: 0.9360 - val_loss: 0.2412 - val_accuracy: 0.9333
Epoch 7/25
198/198 [==============================] - 13s 63ms/step - loss: 0.1146 - accuracy: 0.9529 - val_loss: 0.3441 - val_accuracy: 0.9333
Epoch 8/25
```

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

```
!tar -zcvf Gesture-based-Radiology-Images.tgz gesture.h5
```

```
gesture.h5
```

```
ls
```

```
drive-download-20221029T044045Z-001/  gesture.h5
Gesture-based-Radiology-Images.tgz    model-bw.json
```

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

```
!pip install watson-machine-learning-client
```

```
Requirement already satisfied: watson-machine-learning-client in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.391)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials={
"url":"https://us-south.ml.cloud.ibm.com",
"apikey":"dLSELLnJpa8ZQeS1xU5c6xqeXA8K6dIXMjH17IQBJLsy"
}
client=APIClient(wml_credentials)
client
```

```
<ibm_watson_machine_learning.client.APIClient at 0x7f1cc4461250>
```

```
def guid_space_name(client,Gesture):
    space=client.spaces.get_details()
    return(next(item for item in space['resources'] if item['entity']['name']==Gesture)['metadata']['id'])
space_uid=guid_space_name(client,'Gesture')
```

```
client.set.default_space(space_uid)
```

```
'SUCCESS'
```

```
client.software_specifications.list(100)
```

```
Output exceeds the size limit. Open the full output data in a text editor
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
```

```
software_space_uid=client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid
```

```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

```
model_details=client.repository.store_model(model='Gesture-based-Radiology-Images.tgz',meta_props={
client.repository.ModelMetaNames.NAME:"CNN Model Building",
client.repository.ModelMetaNames.TYPE:'tensorflow_2.7',
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

```
model_id=client.repository.get_model_id(model_details)
model_id
```

```
'88cad567-a954-4bc8-93c0-0ec586942a86'
```

```
client.repository.download(model_id,'gestureproject.tar.gb')
```

```
Successfully saved model content to file: 'gestureproject.tar.gb'

'/home/wsuser/work/gestureproject.tar.gb'
```

OUTPUT:

HOME PAGE:

INTRODUCTION PAGE:

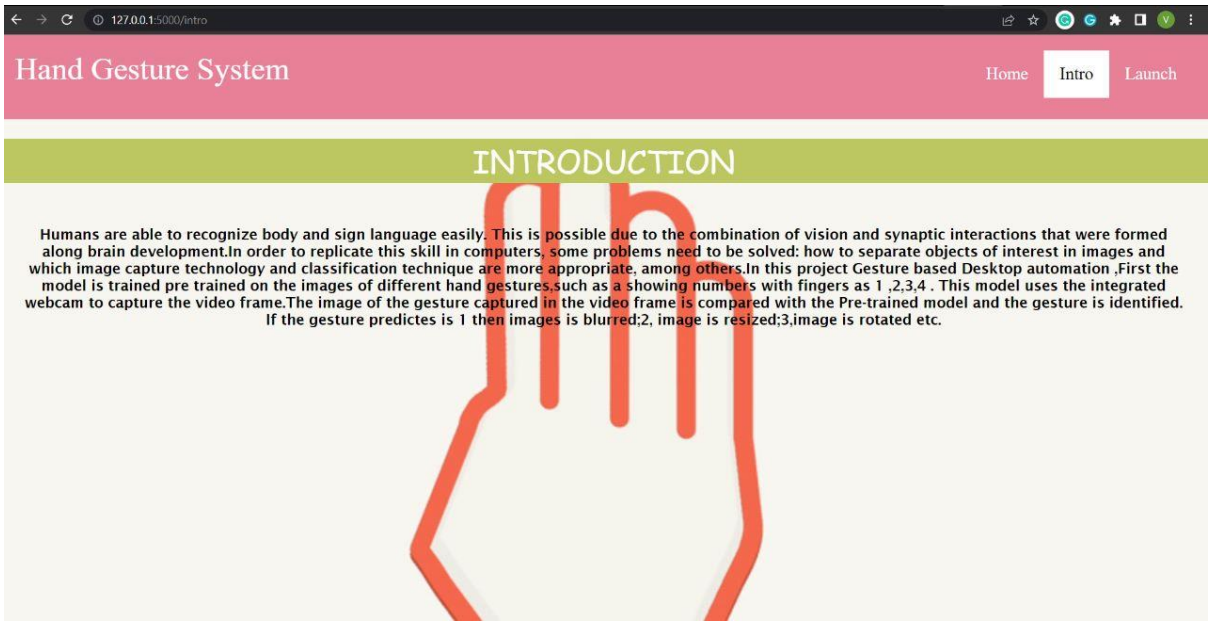
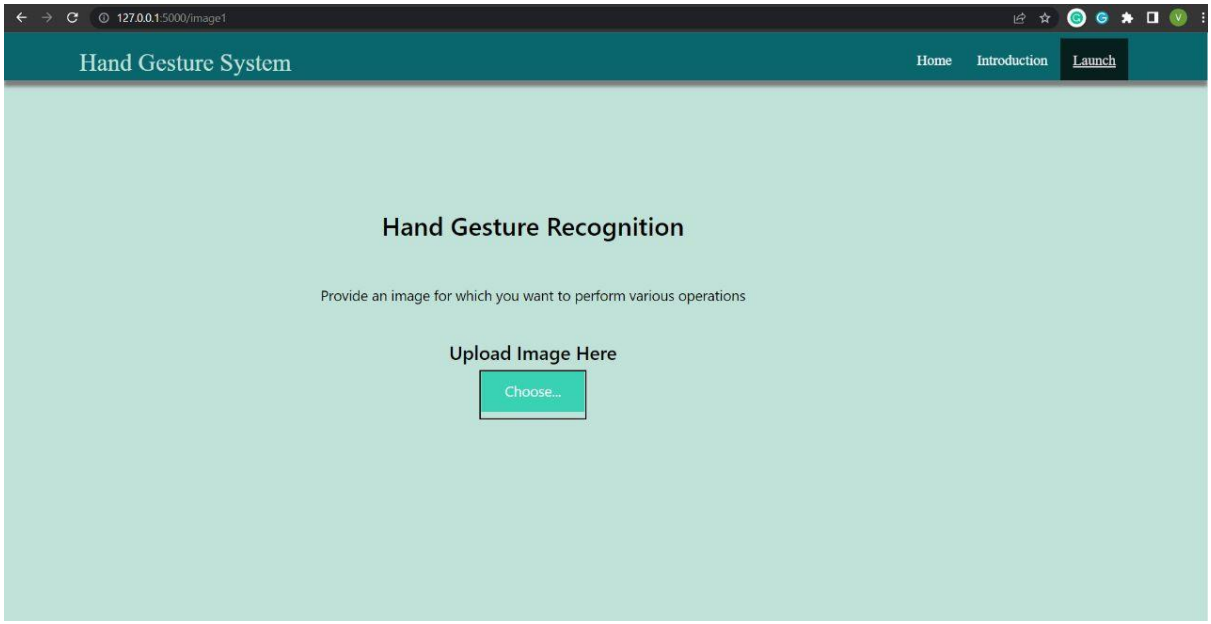
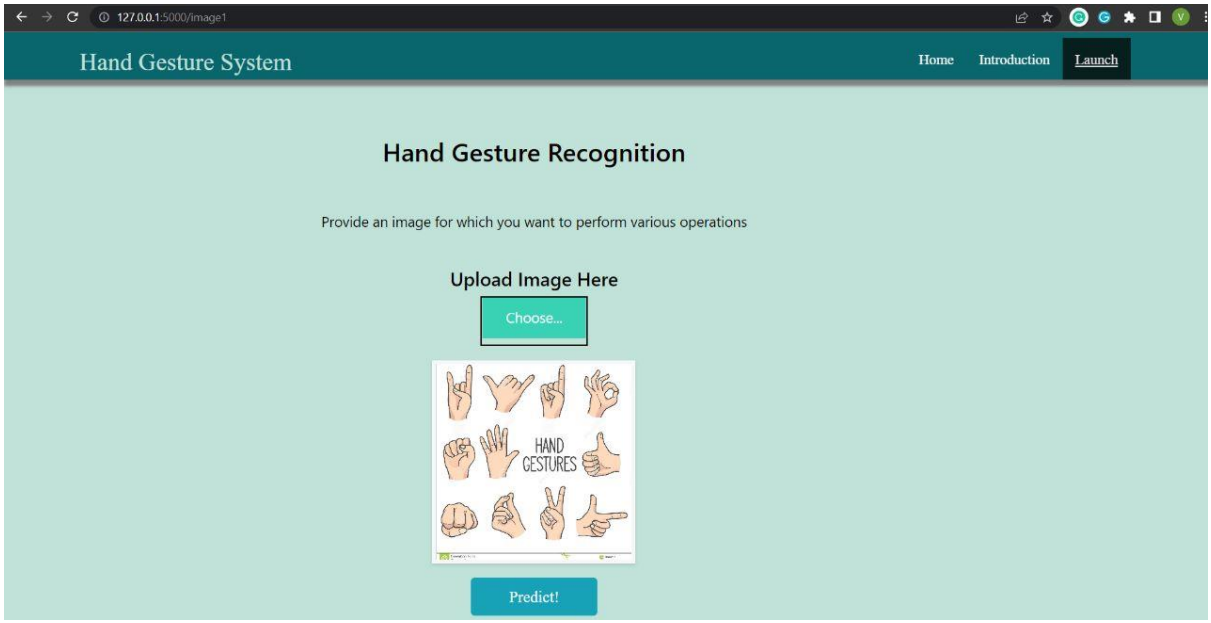


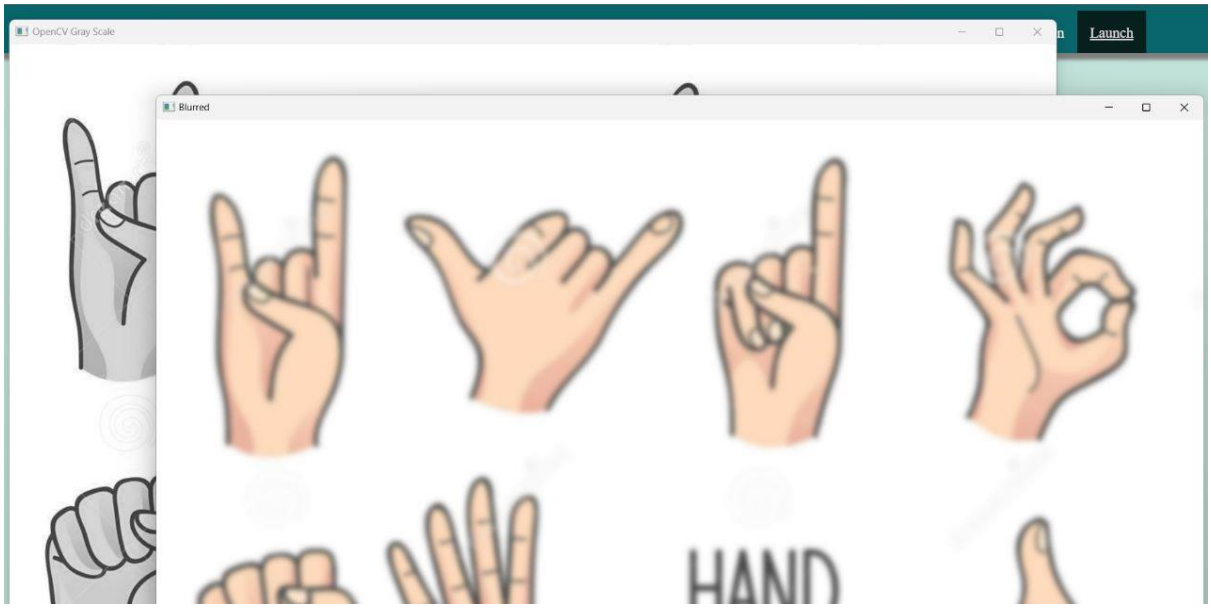
IMAGE UPLOAD PAGE:

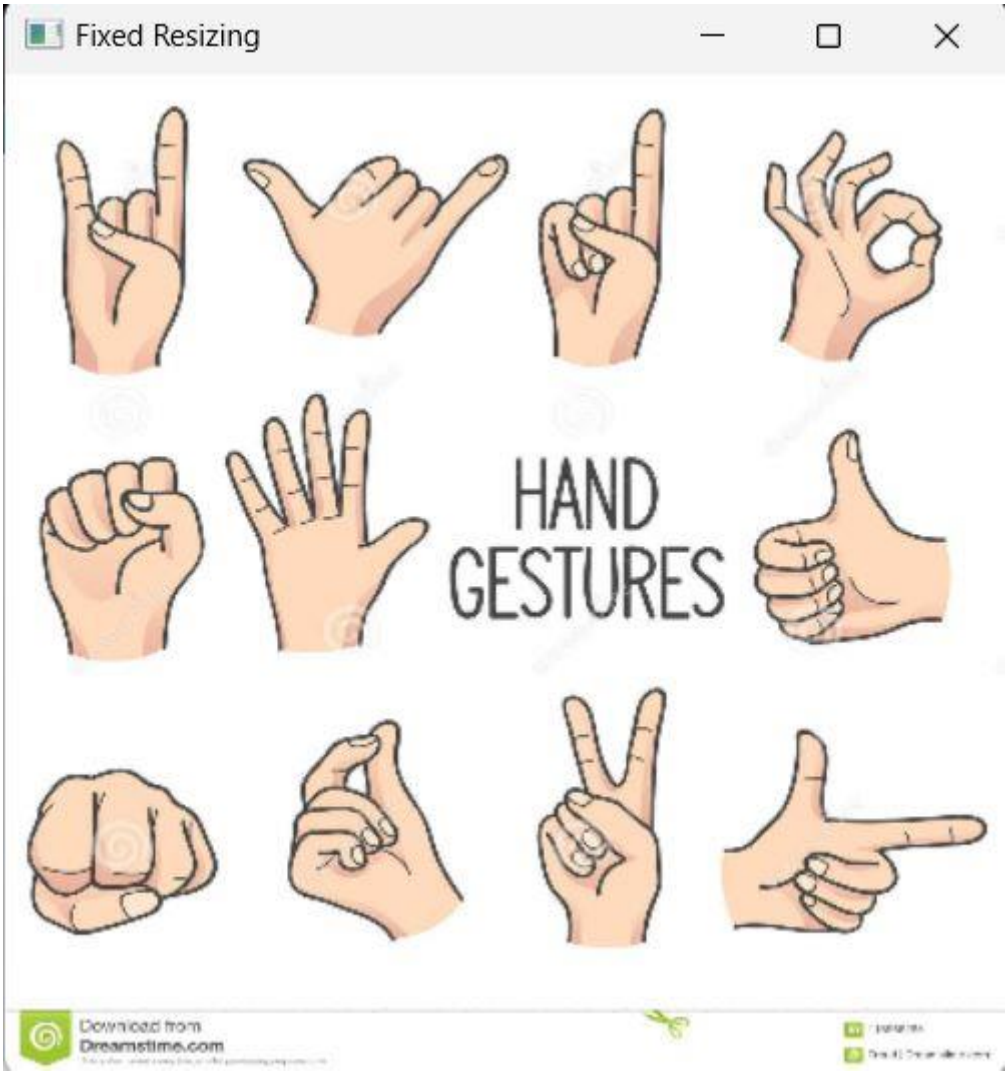


AFTER IMAGE UPLOADED:



MODIFED IMAGES AFTER USING HAND GESTURES:




Fixed Resizing
HAND
GESTURES
Download from
Dreamstime.com

A.2 GitHub & PROJECT DEMO LINK

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-9361-1658997954

Project Demo Link

https://www.youtube.com/watch?v=a4Fgl91z2uA