PROJECT REPORT

ARTIFICIAL INTELLIGENCE

A GESTURE BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

TEAM ID : PNT2022TMID39604

TEAM LEADER - SHABNAM BANU H - 510619104062

TEAM MEMBERS

NARMATHA P - 510619104051

RAJESHWARI R - 510619104055

SASI REKA A - 510619104061

SUPRIYA P - 510619104070

VAISHALI S - 510619104075

Of

BACHELOR OF ENGINEERING

COMPUTER SCIENCE AND ENGINEERING

C.ABDUL HAKEEM COLLEGE OF ENGINEERING AND TECHNOLOGY

ANNA UNIVERSITY::CHENNAI 600 025.

CONTENT

1. INTRODUCTION

1.1 Project Overview

In this project we use gestures to browse images obtained during radiology. Gestures refer to non verbal form of communication made using hands. A major challenge involved in this process is to provide doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work. Keyboards and pointing devices, such as a mouse, are today's principal method of human-computer interaction. However, the use of computer keyboards and mice by doctors and nurses in intensive care units (ICUs) is a common method for spreading infections. Humans can recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development. In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others. In this project Gesture based Desktop automation, First the model is trained pre trained on the images of different hand gestures, such as a showing numbers with fingers as 1,2,3,4. This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the gesture predicts is 0 - then images is converted into rectangle, 1 - image is Re-sized into (200,200), 2 - image is rotated by -45°, 3 - image is blurred, 4 - image is Re-sized into (400,400), 5 - image is converted into grayscale etc.

1.2 Purpose

After uploading the image, our portal uses the integrated webcam to capture the video frame using OpenCV. The gesture captured in the video frame is compared with the pre-trained model and the gesture is identified.

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.
- How-ever 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

- A Gesture-based Tool for Sterile Browsing of Radiology Images - research paper by national library of medicine

- Robust Part-Based Hand Gesture Recognition Using Kinect SensorZhou Ren,Junsong Yuan, Member, IEEE, Jingjing Meng, Member, IEEE, and Zhengyo Zhang, Fellow, IEEE, 15, AUGUST 2013.

- 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

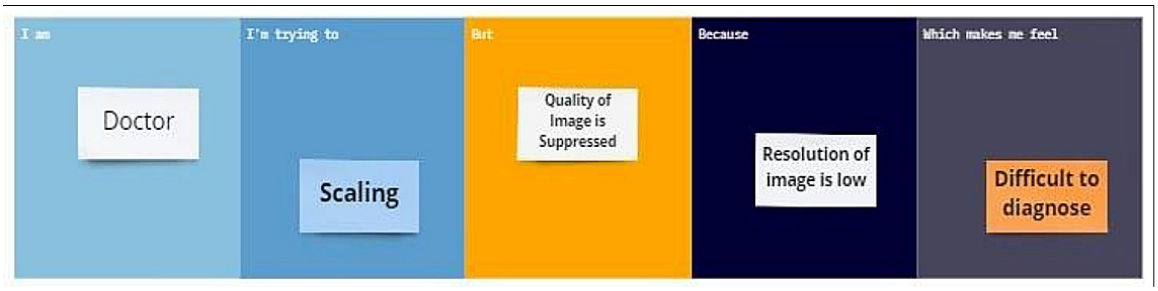
- Intension, Context and Gesture Recognition for Sterile MRI Navigation in the Operating Room by Agency for Healthcare Research and Quality (AHRQ)

- 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.

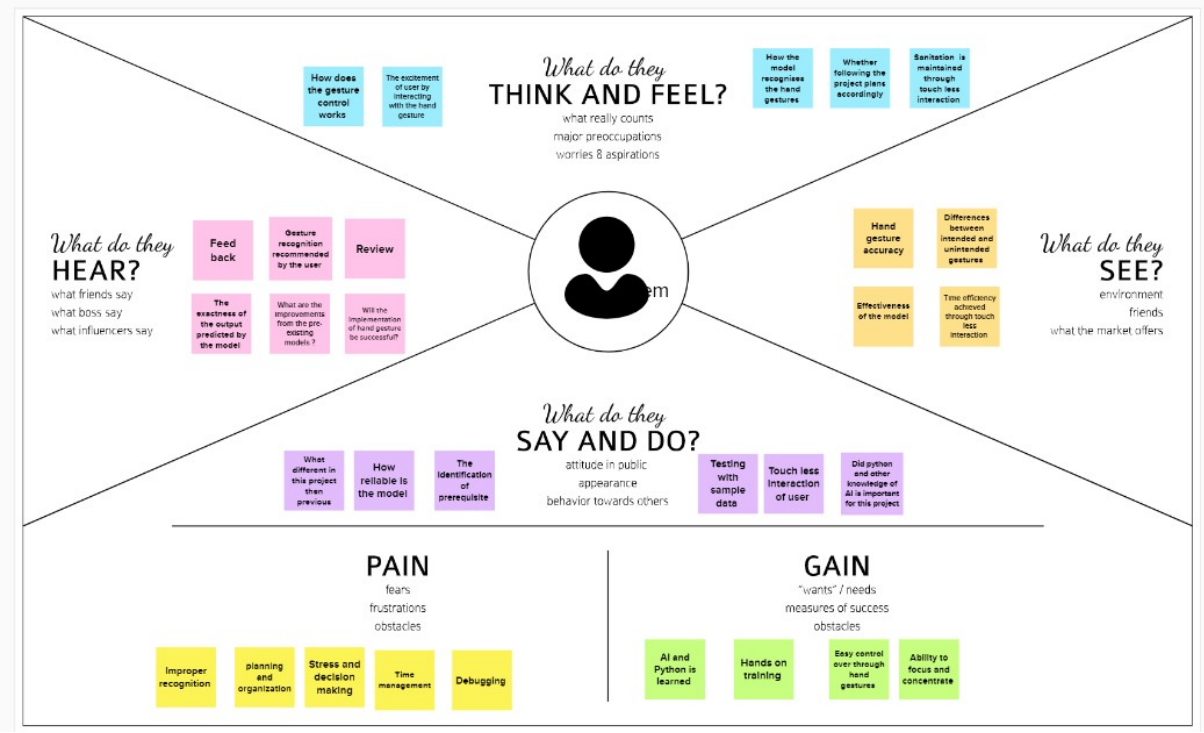
2.3 Problem Statement Definition

A major challenge involved is to provide Doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work. How-ever 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.



3. IDEATION & PROPOSED SOLUTION

3.1 Empathy-Map-Canvas



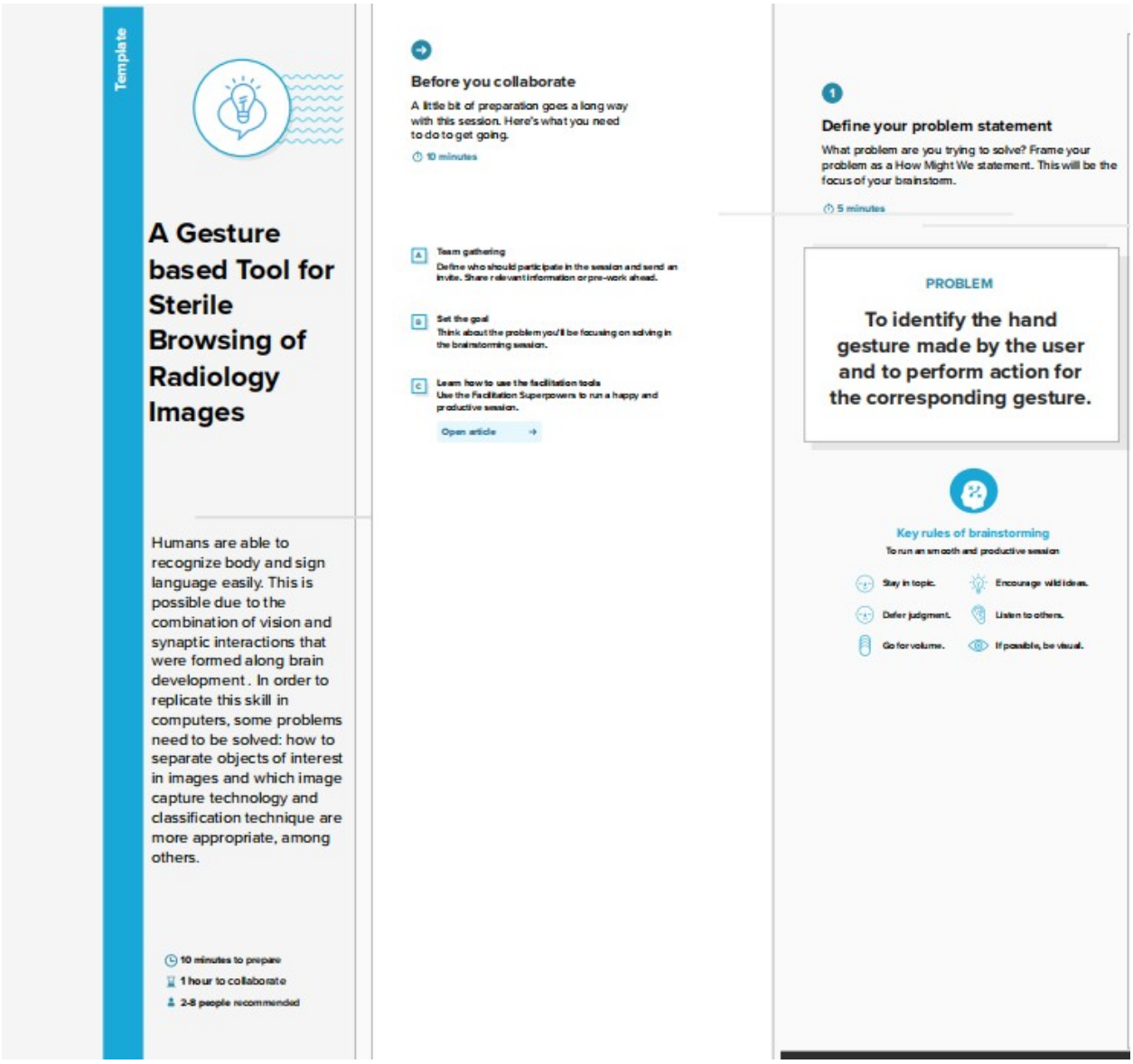
3.2 Ideation and BrainStorming

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.

Step3: Idea-Prioritization.


Template
A Gesture based Tool for Sterile Browsing of Radiology Images
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.
10 minutes to prepare
1 hour to collaborate
2-8 people recommended
Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
10 minutes
A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.
C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.
Open article
1
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes
PROBLEM
To identify the hand gesture made by the user and to perform action for the corresponding gesture.
Key rules of brainstorming
To run an smooth and productive session
Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

2

Brainstorm

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

10 minutes

SHABNAM BANU H

- Simple user interface
- Identify the gesture
- Difference intentional and unintentional gesture
- Hand gesture interpretation

SUPRIYA P

- Interactive user interface to upload image
- 3D recognition of gesture
- To identify exact action for same type of gestures
- Reducing low false positive rates

VAISHALI S

- Most effective algorithm to track image
- Quick analysis of radiology images
- Preservation of radiology images
- Train and test the model with maximum data

NARMATHA P

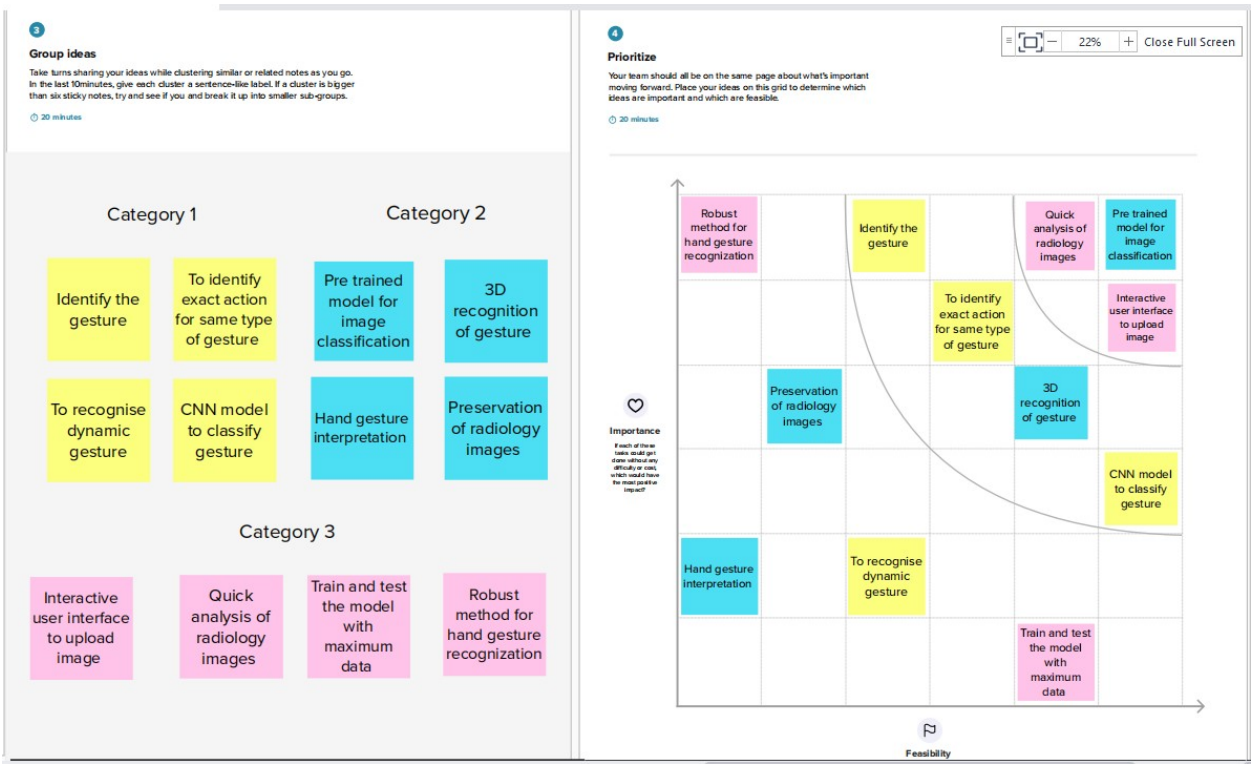
- Pretrained model for image classification
- To recognise dynamic gesture
- Quick expressing of message
- To detect hand click ons accurately

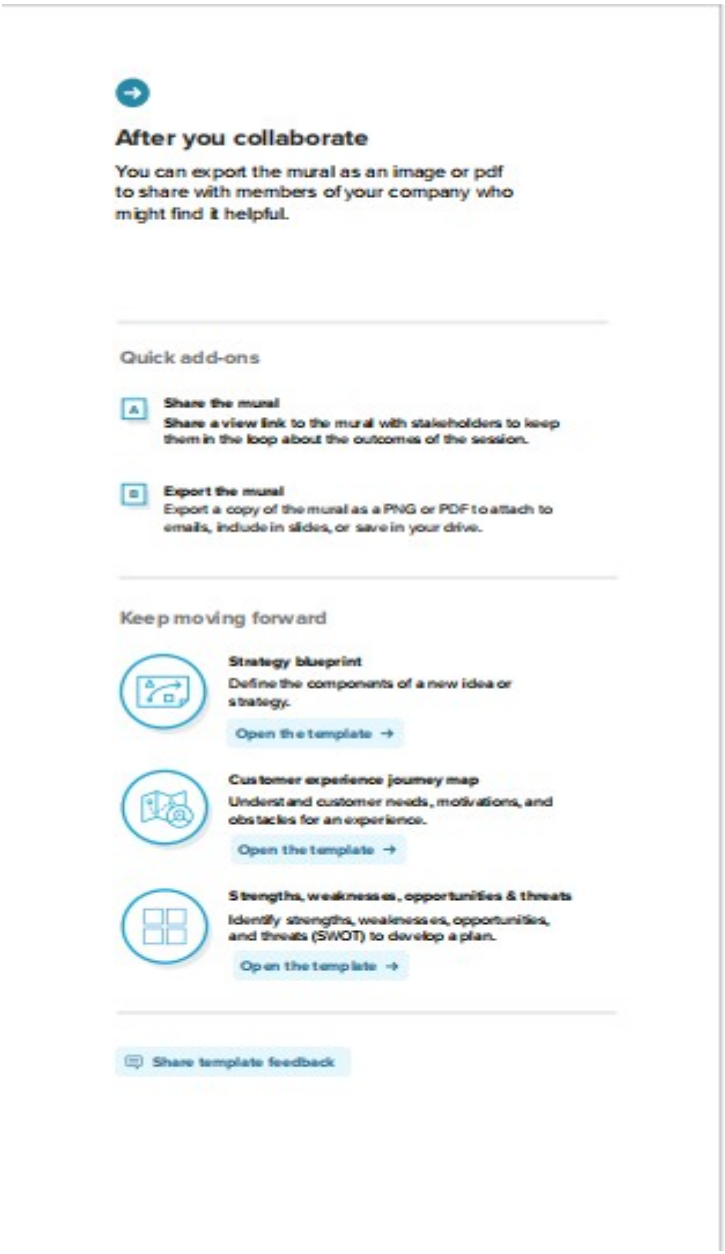
SASI REKA A

- User friendly website
- CNN model to classify gesture
- Touch less user interface
- Navigate in virtual environment

RAJESHWARI R

- Time saving
- Maintaining sterilization on operation theatre
- Robust method for hand gesture recognization
- Extra gestures for more operations


3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
Category 1
Category 2
Identify the gesture
To identify exact action for same type of gesture
Pre trained model for image classification
3D recognition of gesture
To recognise dynamic gesture
CNN model to classify gesture
Hand gesture interpretation
Preservation of radiology images
Category 3
Interactive user interface to upload image
Quick analysis of radiology images
Train and test the model with maximum data
Robust method for hand gesture recognization
4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
22%
Close Full Screen
Robust method for hand gesture recognization
Identify the gesture
Quick analysis of radiology images
Pre trained model for image classification
To identify exact action for same type of gesture
Interactive user interface to upload image
Preservation of radiology images
3D recognition of gesture
CNN model to classify gesture
Hand gesture interpretation
To recognise dynamic gesture
Train and test the model with maximum data
Importance
Feasibility



3.3 Proposed Solution:

- The interaction with intervention 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.

- 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.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To avoid bacterial infection and to analyse radiology images using hand gesture.
2.	Idea / Solution description	Using artificial intelligence to assist doctors by taking hand gestures as input and perform necessary actions on radiology images
3.	Novelty / Uniqueness	The gesture helps to manipulate images and doctors to stay focused.
4.	Social Impact / Customer Satisfaction	The proposed solution provide a good manipulation of radiology images for surgeon during surgery and helps in maintaining focus and faster response time.
5.	Business Model (Revenue Model)	A hand based gesture recognition can detect any type of gesture given as input with pre defined dataset.
6.	Scalability of the Solution	The proposed solution allows learning of new gesture with predefined dataset and effective training.

3.4 Problem Solution fit

1. CUSTOMER SEGMENT(S)	6.CUSTOMER STATE LIMITATION	5. AVAILABLE SOLUTION
- Doctor - Patients	- Available device - Limited customer interaction - Network connection - Spending power - Inconsistent to detect & classify gesture - Faster gesture interaction	- Real time analytics - Hand based analysis - Vision based analysis - Maintain proper sterilization - Image processing and classification - Identify gestures

2. PROBLEMS/PAINS	9. ROOT/CAUSE	7. BEHAVIOUR
- Awareness - Wearable devices on user's body - Expensive camera - Distance from doctor and camera - Communication is complex	- Diversity of gesture due to high tolerance of error - Background noise - Camera's resolution and quality - Fighting the lag - Doctors unavailable	- Improper care by doctors towards patient - Low motion gesture challenging - Sensors are required - Good internet facility

3. TRIGERS TO ACT	10. YOUR SOLUTION	8. CHANNELS OF BEHAVIOUR
- Doctor to make operation easier - Time saving - Faster result to patient - Faster results to patients while performing operation **4. EMOTIONS** Before - Fear - Stress After - Smart work of doctor - Patients feel very easy	- Make AI based machine that can diagnose using images (X-ray, Scan) can reduce the work of doctors. - To make simple UI website to recognise hand gesture using radiology images. This reduces the economic expenditure	Online - Advertising in social media - Using in industries for in health case, automobiles in electronic devices etc. - Online consulting of patients Offline - Offline consultation is required for patients who has severe health issues - Guidance to Patient.

4. REQUIREMENT ANALYSIS:

4.1. Functional Requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Hand detection	Filters out hand from video capturing device.
FR-2	Skin detection	Filter the objects of non skin-coloured.
FR-3	Filtered object detection	Read and recognize clusters skin coloured objects.
FR-4	Hand calibration	Perform adjustment according to user's dominant hand.
FR-5	Browsing gesture control mode	Hand gesture recognized for commands.

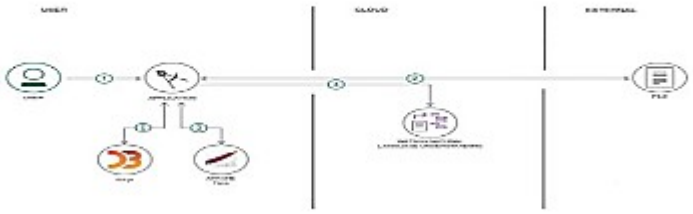
4.2 Non-Functional Requirement:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Easy to use for all users with minimal instruction. It is understandable by non-technical users.
NFR-2	**Security**	Application will be permissible to used only in secure Networks and less feasibility of insecurity.
NFR-3	**Reliability**	It is operable in all lighting conditions. Regardless, of the brightness level in user's operating environment.
NFR-4	**Performance**	Minimize the number of calculation to perform hand gesture detection and to improve image quality resolution.
NFR-5	**Availability**	This application supports recognition of gesture only through internet connection.
NFR-6	**Scalability**	Scalable to support future developments and shall be at least 50% extensible to allow new gesture recognition features to be added to the systems.

5. PROJECT DESIGN:

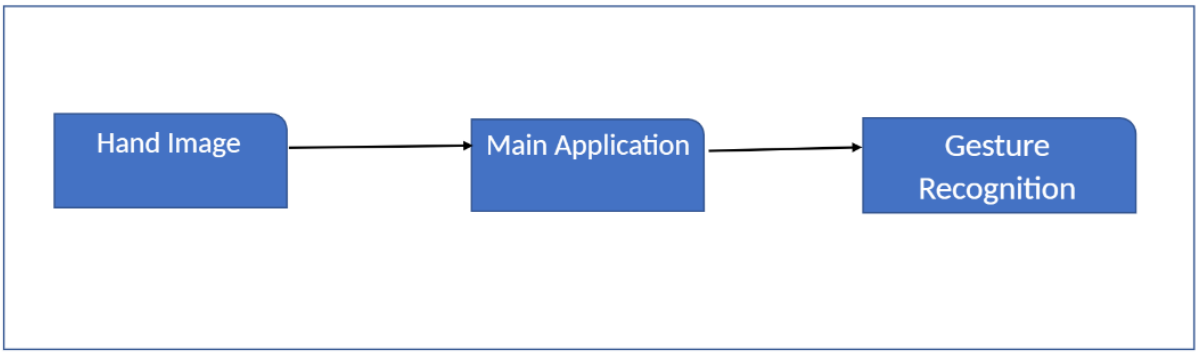
5.1 Data Flow Diagrams:

Flow

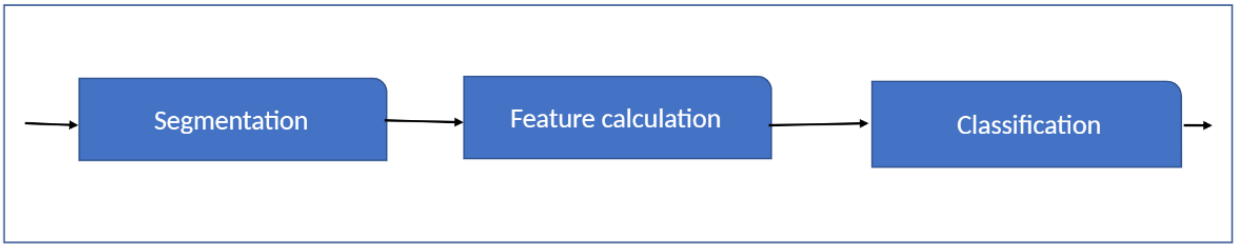


1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.

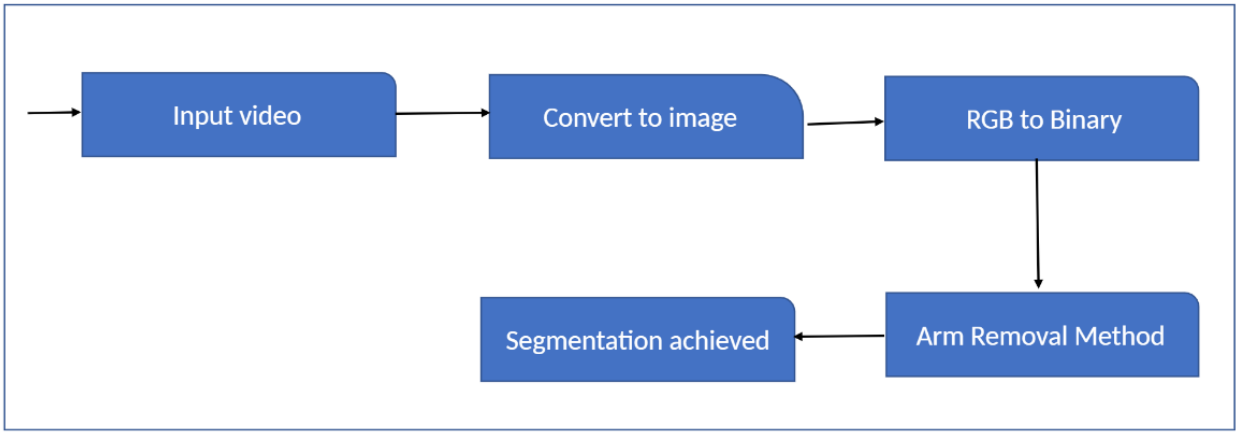
DFD Level 0



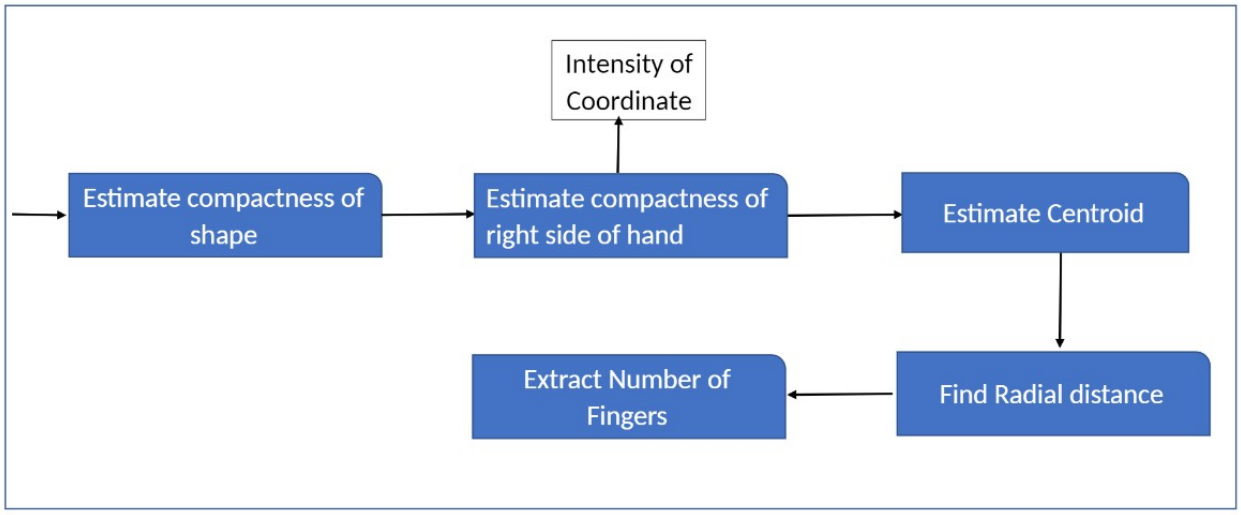
DFD Level 1



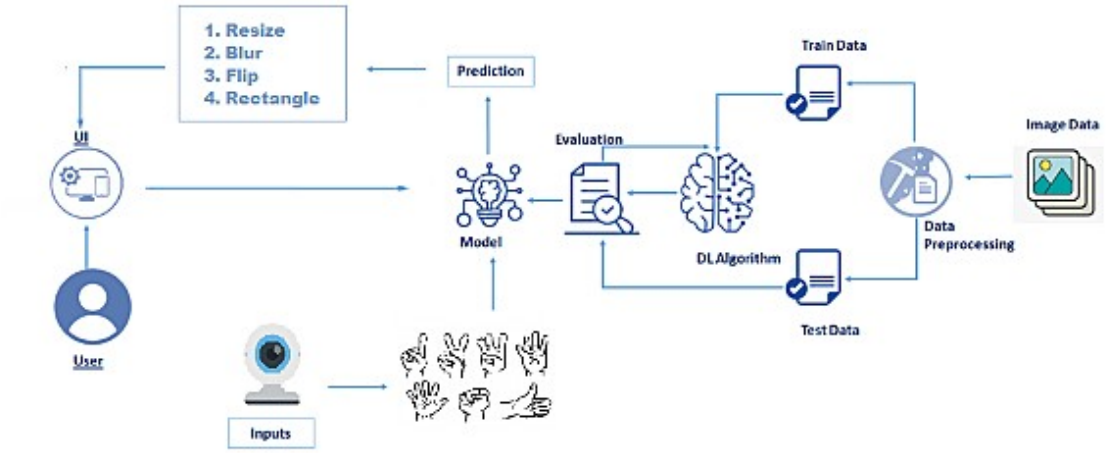
DFD Level 2



DFD Level 3



5.2 Solution and Technical Architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Launch	USN-1	As a user I can launch to the further web page To upload image	I can access the webpage	High	Sprint-4
		USN-2	As a user I can use chrome/fire fox/edge	Different access of web browsers	High	Sprint-1
Customer care executive	Availability	USN-1	As a user when the website is Unresponsive or any internal errors	Webpage not responding	Medium	Sprint-4
		USN-2	Identifying gesture inaccurately	Webcam detection	Medium	Sprint-5
Administrator	IBM cloud	USN-1	Access the database	Database Management	High	Sprint-3
		USN-2	Server Crash	Resolve the errors/issue	Medium	Sprint-5

6. PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning and Estimation:

1.Collect the Image Data.

2.Pre-process the collected images.

3.Train the model

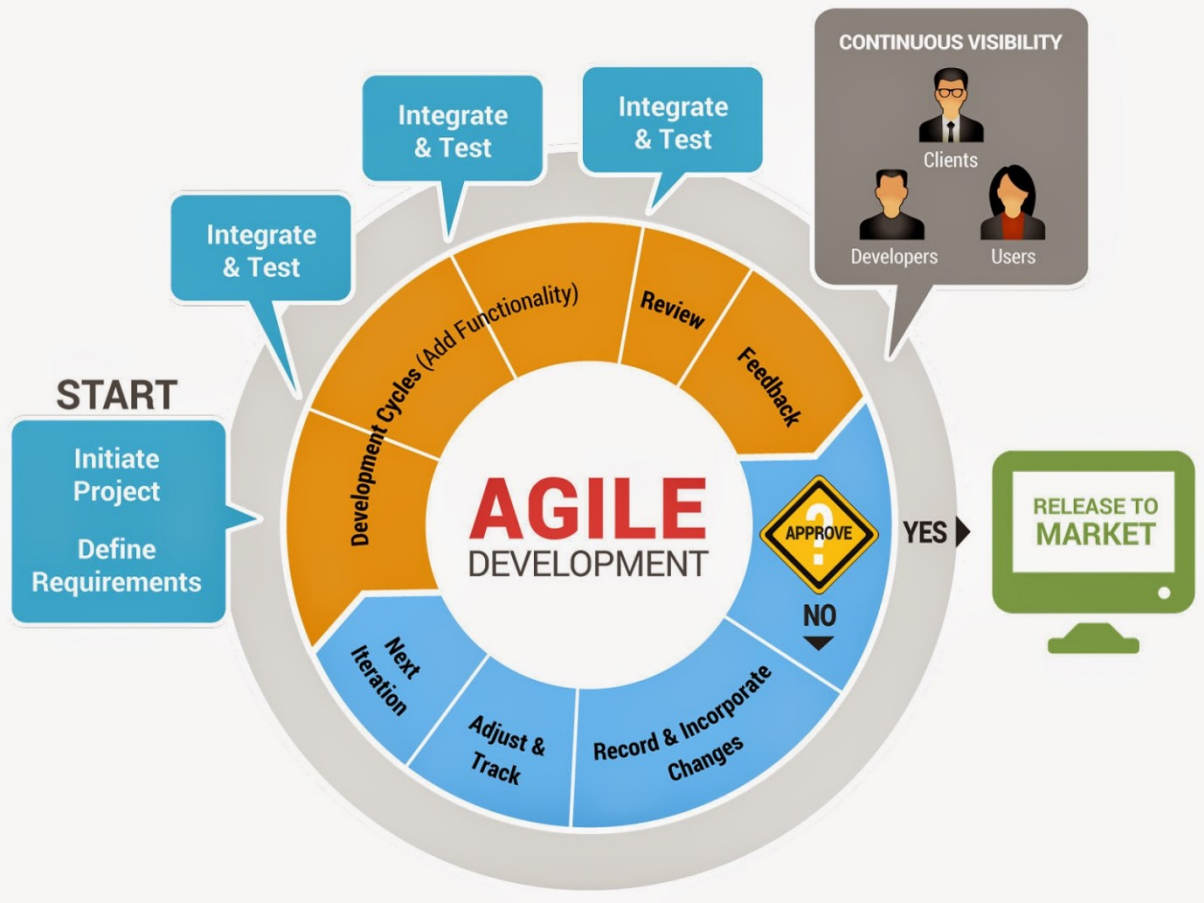
4.Test the model.

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 web pages.



6.2 Sprint Delivery Schedule:

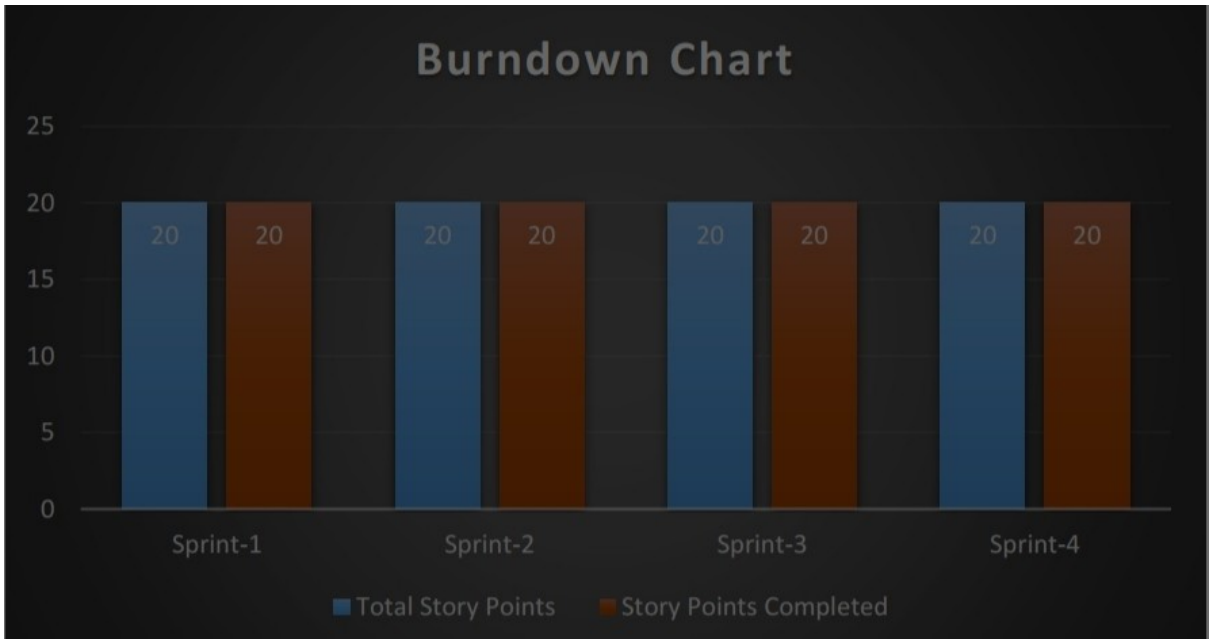
Product Backlog , Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data collection (Dataset)	USN-1	As a user, I will download dataset of gestures for this project.	2	High	Shabnam Banu H Sasi Reka A Supriya P Narmatha P Rajeshwari R Vaishali S
Sprint-1	Image Preprocessing	USN-2	As a user, I will import necessary libraries for image data generator and configure the image data generator class.	2	High	Shabnam Banu H Supriya P Vaishali S
Sprint-1	Image Preprocessing	USN-3	As a user, I will train and test the dataset to apply image data generator functionality.	2	High	Shabnam Banu H Supriya P Vaishali S
Sprint-2	Model building	USN-4	As a user, I can import necessary libraries and initialize the model.	2	Low	Shabnam Banu H Supriya P Vaishali S
Sprint-2	Model Building	USN-5	As a user, I will add CNN layers , Dense layers And configure the learning process.	2	Low	Shabnam Banu H

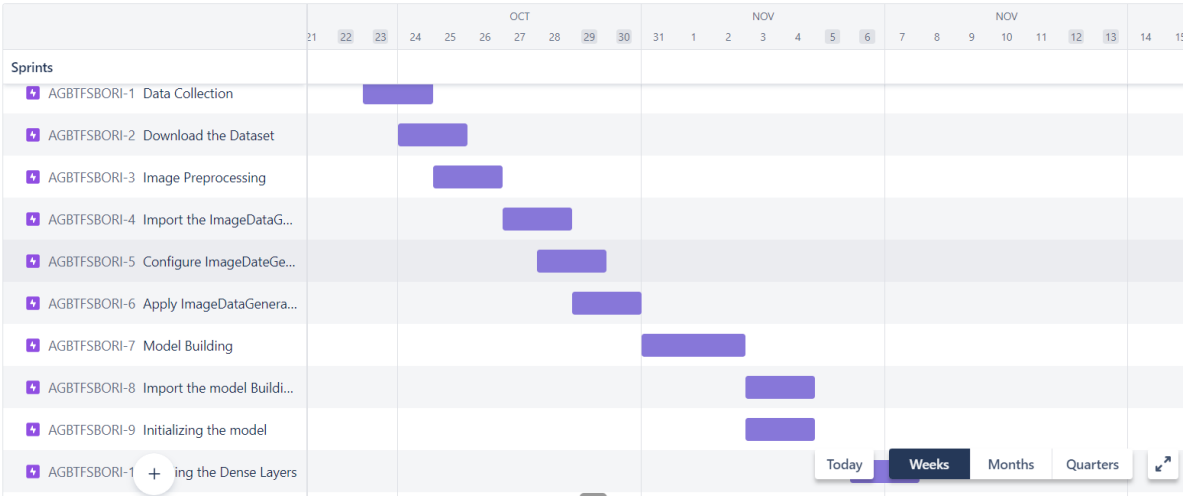
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
						Supriya P Vaishali S
Sprint-2	Model Building	USN-6	As a user, I will train, save and test the model.	2	Medium	Shabnam Banu H Supriya P Vaishali S
Sprint-3	Application Building	USN-7	As a user, I create html front page (CSS for styling webpage and JS to connect back end).	1	High	Sasi Reka A Narmatha P Rajeshwari R
Sprint-3	Application Building	USN-8	As a user, I use python flask for building back end(for server side scripting).	2	High	Shabnam Banu H Supriya P Vaishali S
Sprint-3	Application Building	USN-10	As a user, going to run the application by combining both front end and back end.	2	High	Shabnam Banu H Sasi Reka A Supriya P Narmatha P Rajeshwari R Vaishali S
Sprint-4	Train the model on IBM	USN-11	As a user, register for IBM cloud.	1	Medium	Shabnam Banu H Sasi Reka A Supriya P Narmatha P Rajeshwari R Vaishali S
Sprint-4	Train the model on IBM	USN-12	As a user, train the model on IBM and integrate it with the flask application.	2	High	Shabnam Banu H Sasi Reka A Supriya P Narmatha P Rajeshwari R Vaishali S

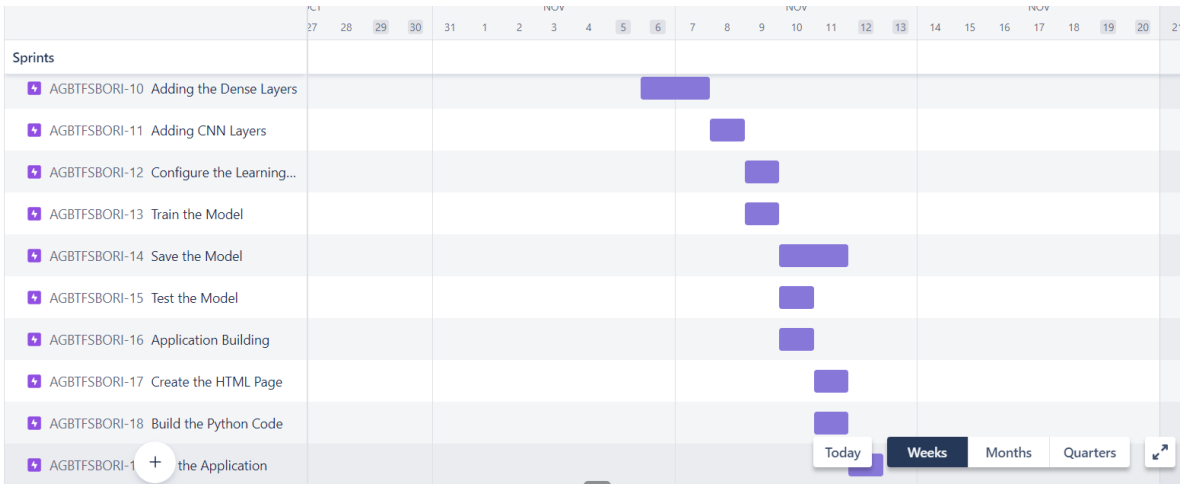
Project Tracker, Velocity & Burndown chart:

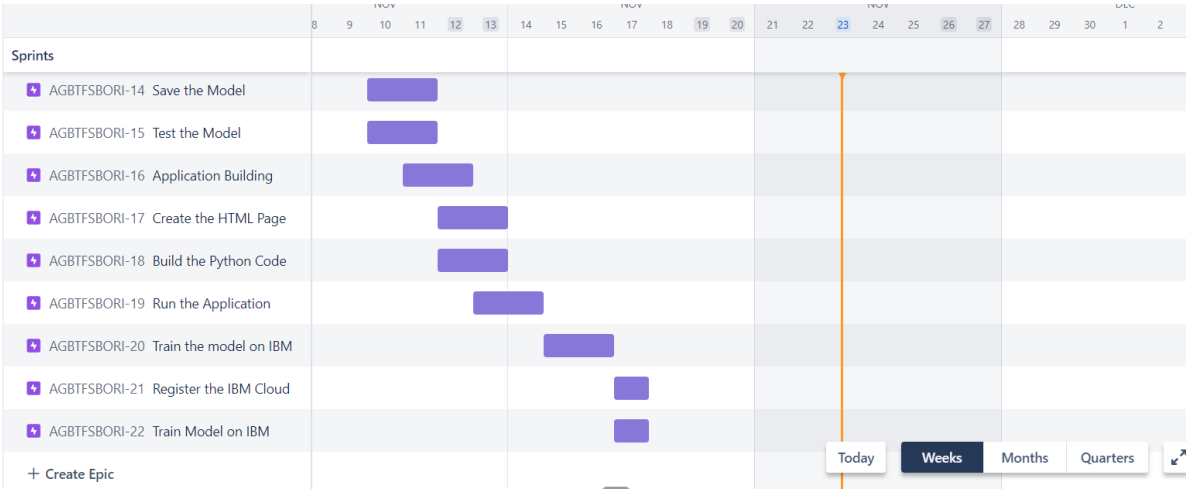
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

#setting parameter for Image Data agumentation to the traing data
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)
```

Apply ImageDataGenerator Functionality to Trainset and Testset

```
Loading our data and performing data agumentation

#performing data agumentation to train data
x_train = train_datagen.flow_from_directory('data/train',target_size=(64, 64),batch_size=5,
                                            color_mode='grayscale',class_mode='categorical')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory('data/test',target_size=(64, 64),batch_size=5,
                                            color_mode='grayscale',class_mode='categorical')

Found 600 images belonging to 6 classes.
Found 30 images belonging to 6 classes.
```

2. MODEL BUILDING

Importing the Model Building Libraries

Importing Neccessary Libraries

```
import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D #Convolutional layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator
```

Initial-zing the Model

```
model=Sequential()
```

Adding CNN Layers

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

# Flattening the layers
classifier.add(Flatten())
```

Adding Dense Layers

```
# Adding a fully connected layer
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=6, activation='softmax')) # softmax for more than 2
```

Configure the Learning Process

Compiling the model

```
# Compiling the CNN
# categorical_crossentropy for more than 2
classifier.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Training Model

Train the model with our image dataset

Fitting the model

```
classifier.fit_generator(
        generator=x_train,steps_per_epoch = len(x_train),
        epochs=20, validation_data=x_test,validation_steps = len(x_test))# No of images in test set
```

Save the Model

Saving our model

```
# Save the model
classifier.save('gesture.h5')

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

Test the Model

Predicting our results

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model = load_model("gesture.h5") #Loading the model for testing
```

```
img = image.load_img(r"E:\PROJECTS\number-sign-recognition\data\test\1\1.jpg",grayscale=True,
                     target_size= (64,64))#Loading of the image
x = image.img_to_array(img)#image to array
x = np.expand_dims(x,axis = 0)#changing the shape
pred = model.predict_classes(x)#predicting the classes
pred
```

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

```
index=['0','1','2','3','4','5']
result=str(index[pred[0]])
result
```

```
'1'
```

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
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 image1():
    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 result

- **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 ROI's of many shapes, such 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.

```
# 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)
```

- **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.

```
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)
```

- **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.

```
#loading an image
image1=cv2.imread(file_path)
if prediction[0][0]=='ONE':

    resized = cv2.resize(image1, (200, 200))
    cv2.imshow("Fixed Resizing", resized)
    key=cv2.waitKey(3000)

    if (key & 0xFF) == ord("1"):
        cv2.destroyWindow("Fixed Resizing")

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

    cv2.rectangle(image1, (480, 170), (650, 420), (0, 0, 255), 2)
    cv2.imshow("Rectangle", image1)
    cv2.waitKey(0)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("0"):
        cv2.destroyWindow("Rectangle")

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

elif prediction[0][0]=='THREE':
    blurred = cv2.GaussianBlur(image1, (11, 11), 0)
    cv2.imshow("Blurred", blurred)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("3"):
        cv2.destroyWindow("Blurred")
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 local host where you can view your web page

```
(base) E:\>cd E:\PROJECTS\number-sign-recognition\Flask

(base) E:\PROJECTS\number-sign-recognition\Flask>python app.py
```

Then it will run on local host:5000

```
* 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 local host (https://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	6	0	0	6
Client Application	45	0	0	45
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	5	0	0	5
Final Report Output	6	0	0	6
Version Control	1	0	0	1

8.2 User Acceptance Testing:

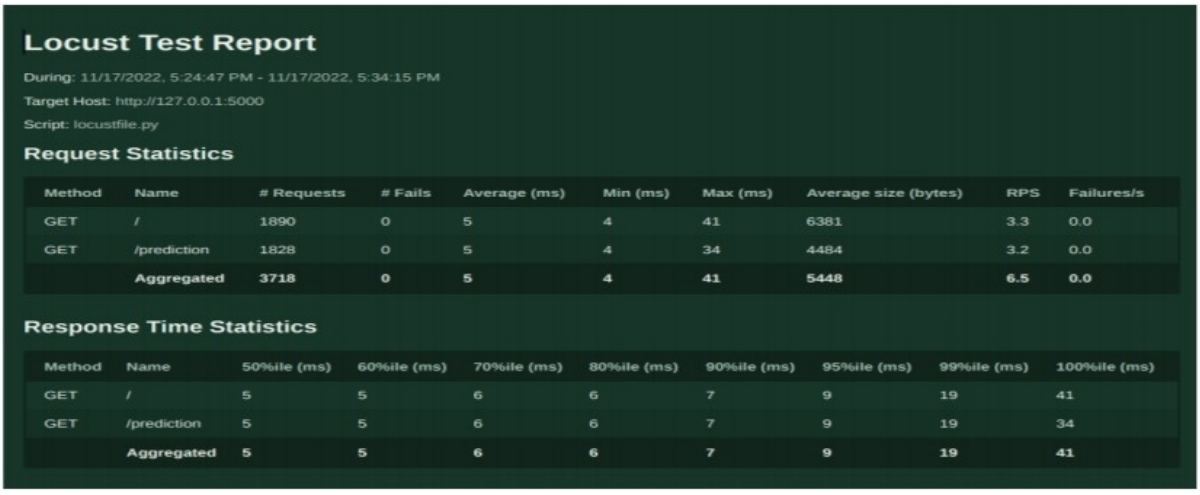
2. Defect Analysis

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	6	4	2	2	14
Duplicate	1	2	1	0	4
External	2	3	0	2	7
Fixed	12	3	2	3	20
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	4	1	1	6
Totals	21	16	8	9	54

9.RESULTS:

9.1 Performance Metrics:



Locust Test Report

During: 11/17/2022, 5:24:47 PM - 11/17/2022, 5:34:15 PM

Target Host: http://127.0.0.1:5000

Script: locustfile.py

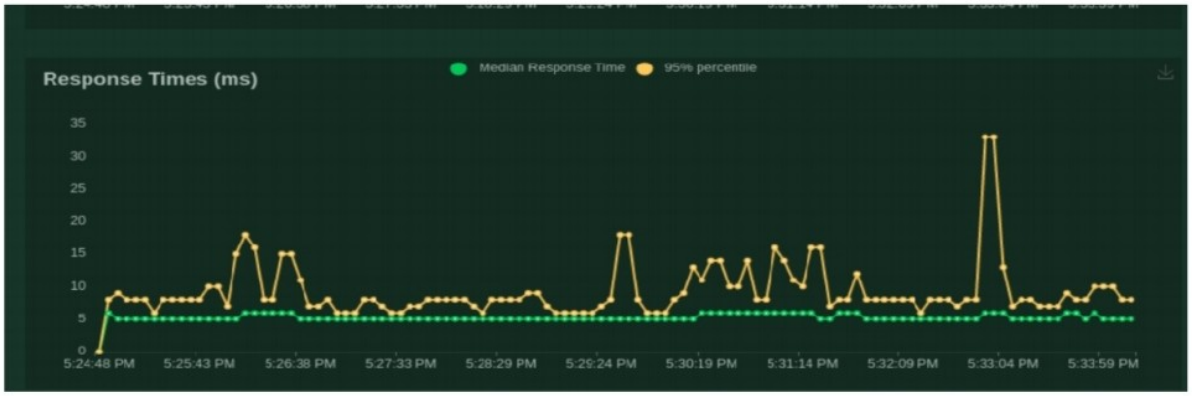
Request Statistics

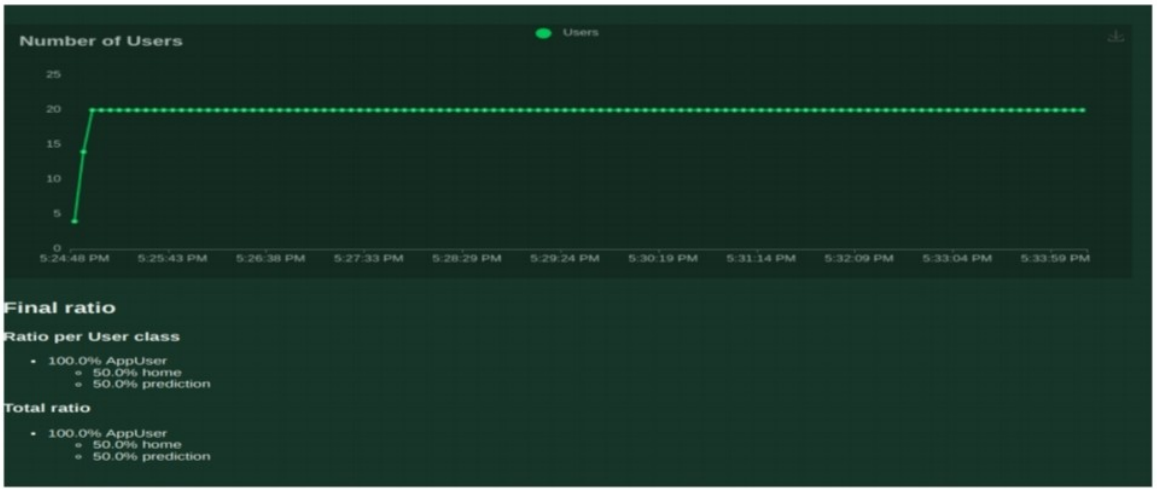
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	1890	0	5	4	41	6381	3.3	0.0
GET	/prediction	1828	0	5	4	34	4484	3.2	0.0
	Aggregated	3718	0	5	4	41	5448	6.5	0.0

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	/	5	5	6	6	7	9	19	41
GET	/prediction	5	5	6	6	7	9	19	34
	Aggregated	5	5	6	6	7	9	19	41

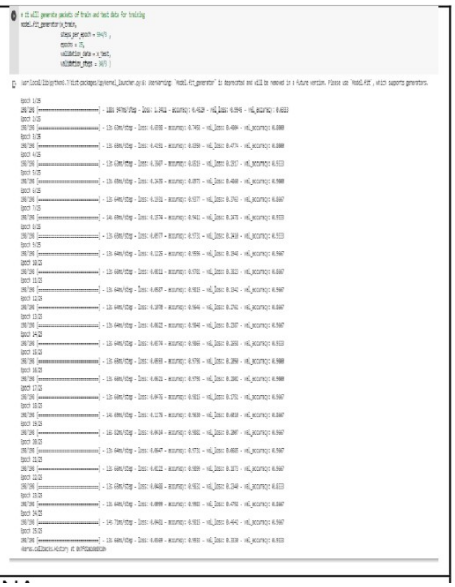

Charts
Total Requests per Second
RPS
Failures/s
8
6
4
2
0
5:24:48 PM
5:25:43 PM
5:26:38 PM
5:27:33 PM
5:28:29 PM
5:29:24 PM
5:30:19 PM
5:31:14 PM
5:32:09 PM
5:33:04 PM
5:33:59 PM


Response Times (ms)
Median Response Time
95% percentile
35
30
25
20
15
10
5
0
5:24:48 PM
5:25:43 PM
5:26:38 PM
5:27:33 PM
5:28:29 PM
5:29:24 PM
5:30:19 PM
5:31:14 PM
5:32:09 PM
5:33:04 PM
5:33:59 PM


Number of Users
Users
25
20
15
10
5
0
5:24:48 PM
5:25:43 PM
5:26:38 PM
5:27:33 PM
5:28:29 PM
5:29:24 PM
5:30:19 PM
5:31:14 PM
5:32:09 PM
5:33:04 PM
5:33:59 PM
Final ratio
Ratio per User class
100.0% AppUser
50.0% home
50.0% prediction
Total ratio
100.0% AppUser
50.0% home
50.0% prediction

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.NO	Parameter	Values	Screen Shot
1.	Model Summary	conv2d (Conv2D) - 320 max_pooling2d (MaxPooling2D) - 0 conv2d_1 (Conv2D) - 9248 max_pooling2d_1 (MaxPooling2D) - 0 flatten (Flatten) - 0 dense (Dense) - 802944 dense_1 (Dense) - 774 ========== Total params: 813,286 Trainable params: 813,286 Non-trainable params: 0	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
2.	Accuracy	Training Accuracy-99% Validation Accuracy-93%	

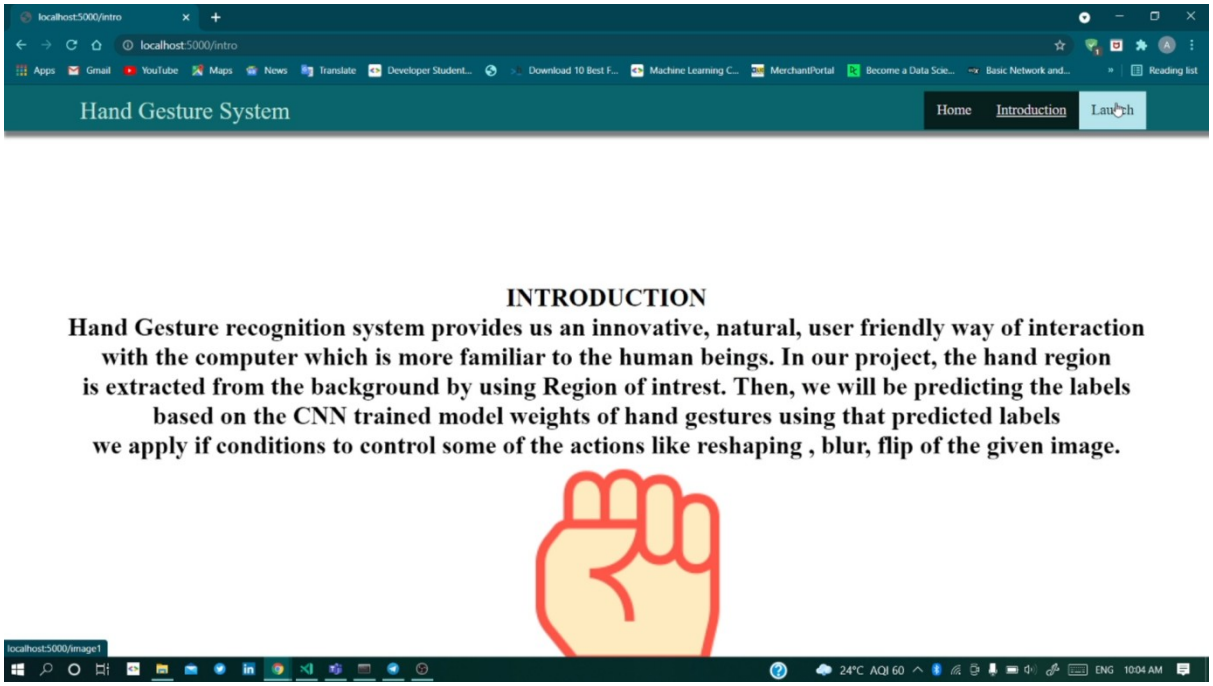
9.2. Output

1: Home Page:

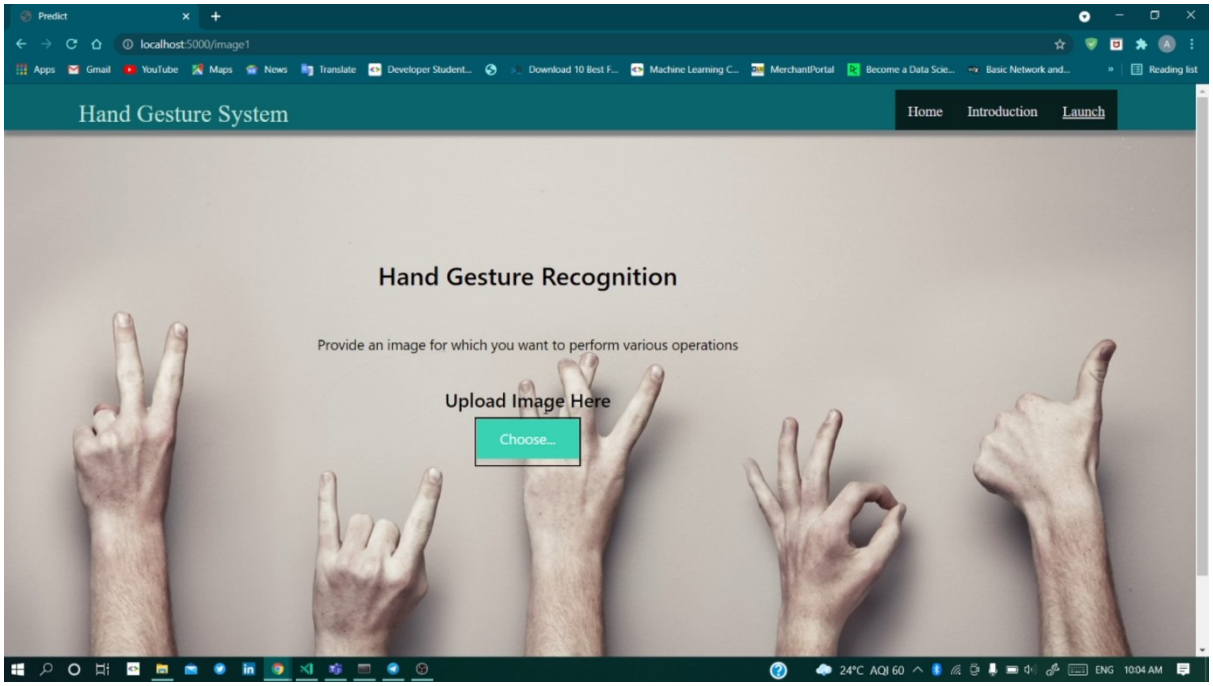
Let's see how our home.html page looks like:

2: Introduction page:

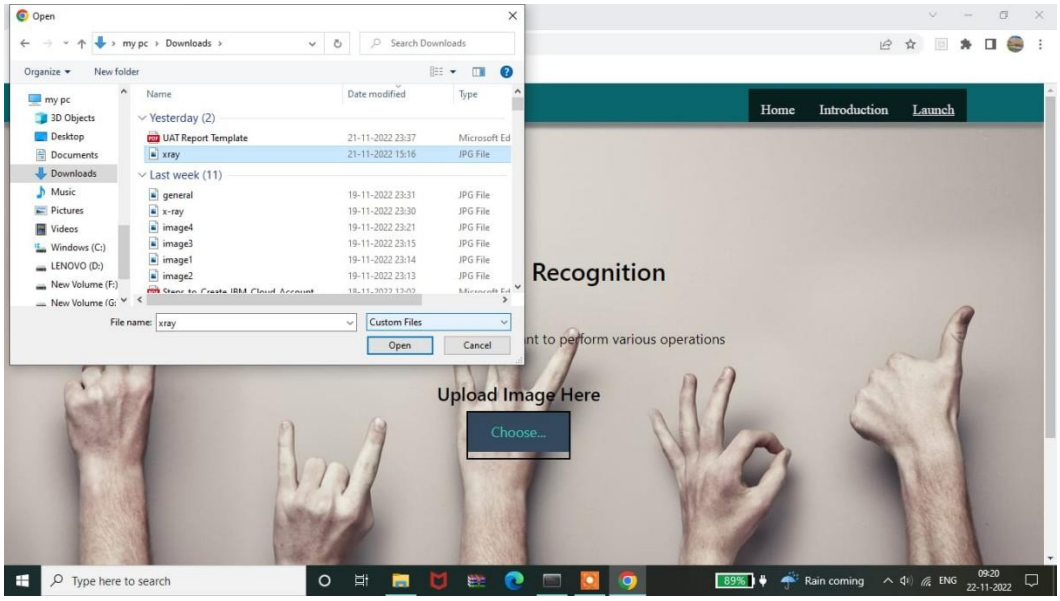
When "intro" button is clicked, local host redirects to "intro.html"



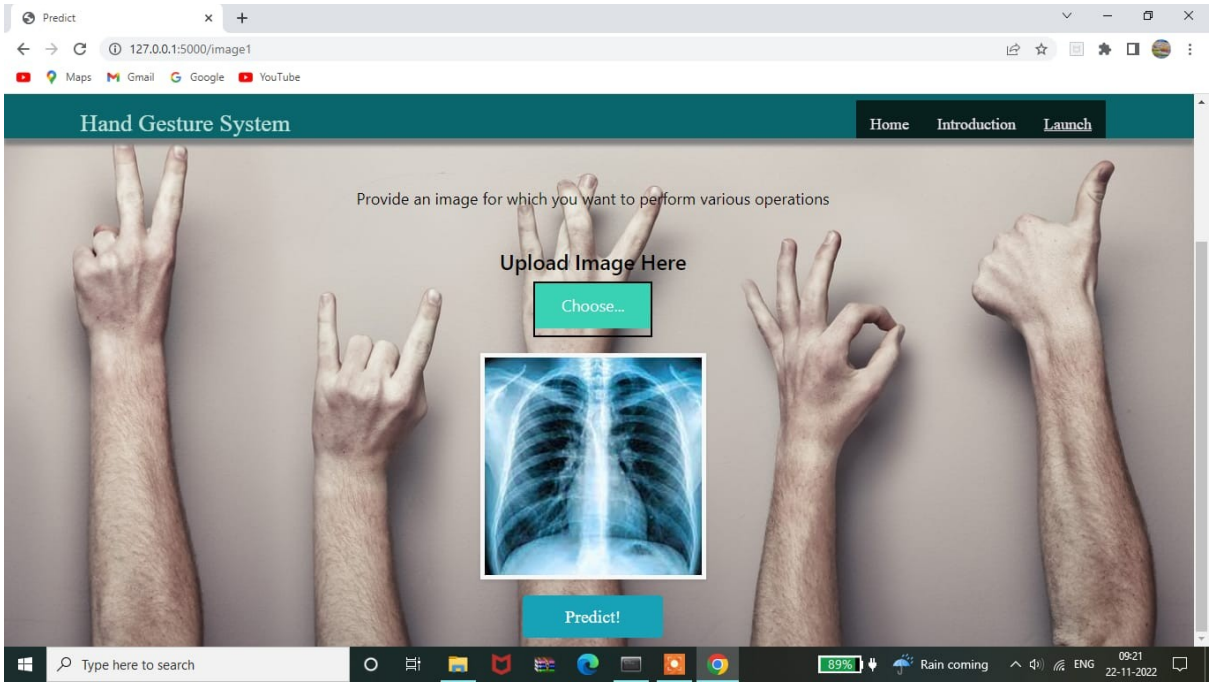
3. Predict Page:



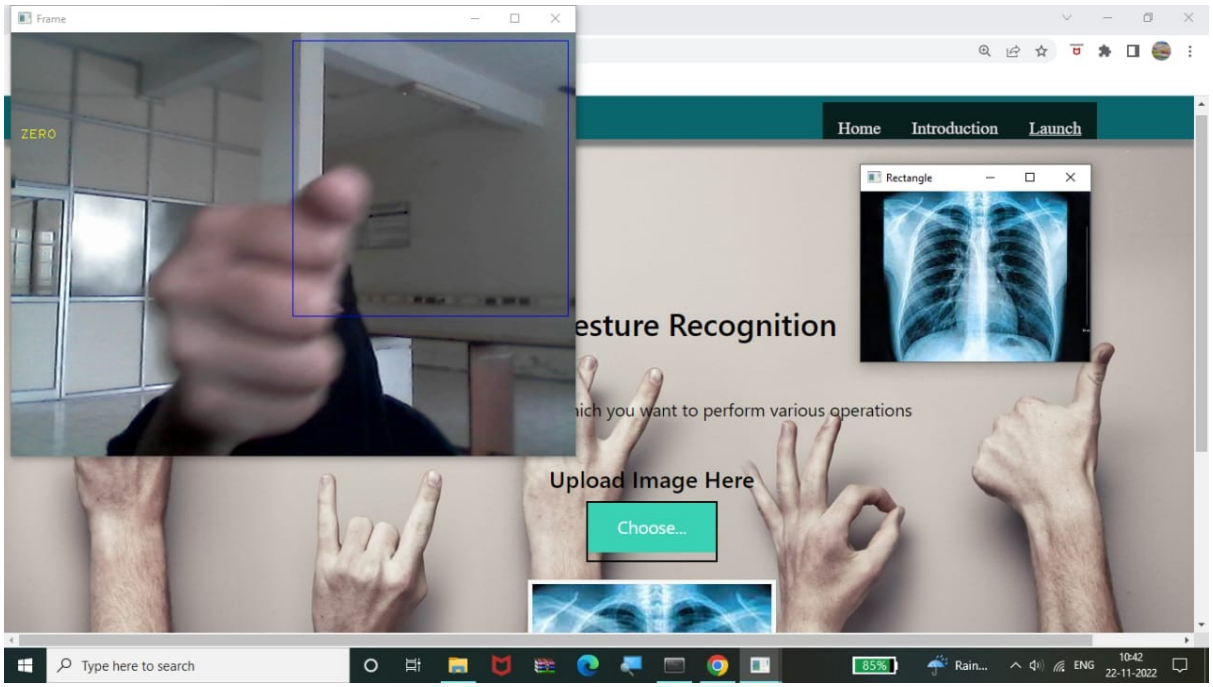
3.1. Upload Image:

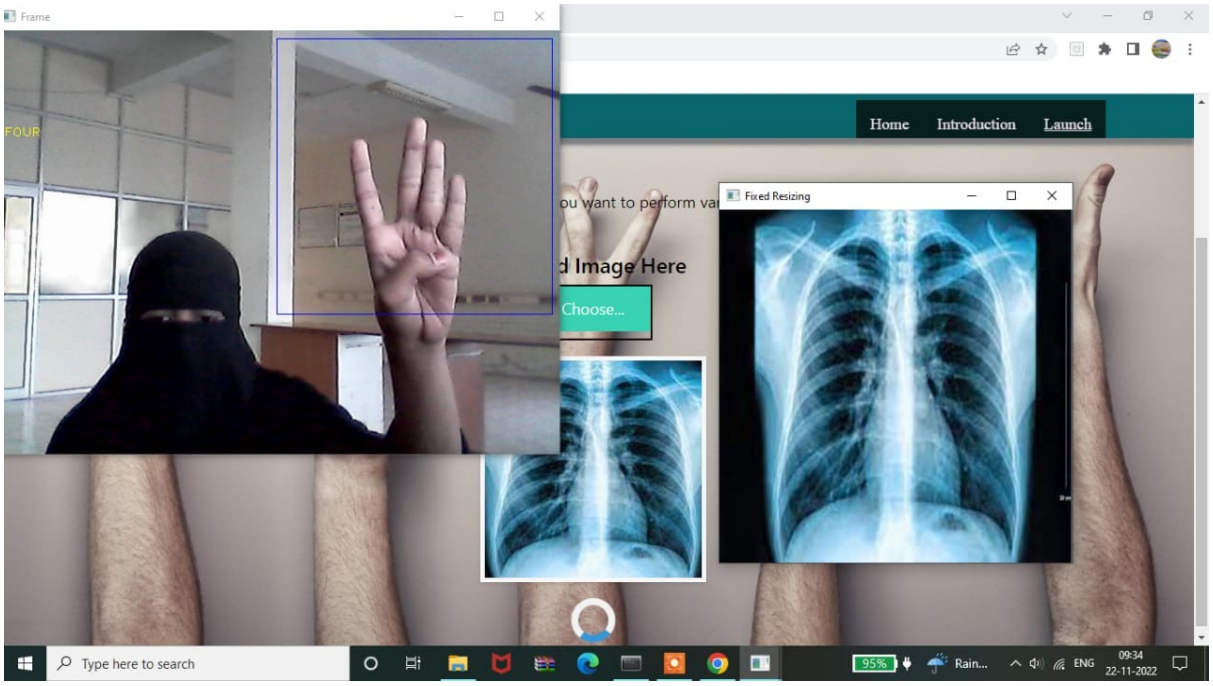


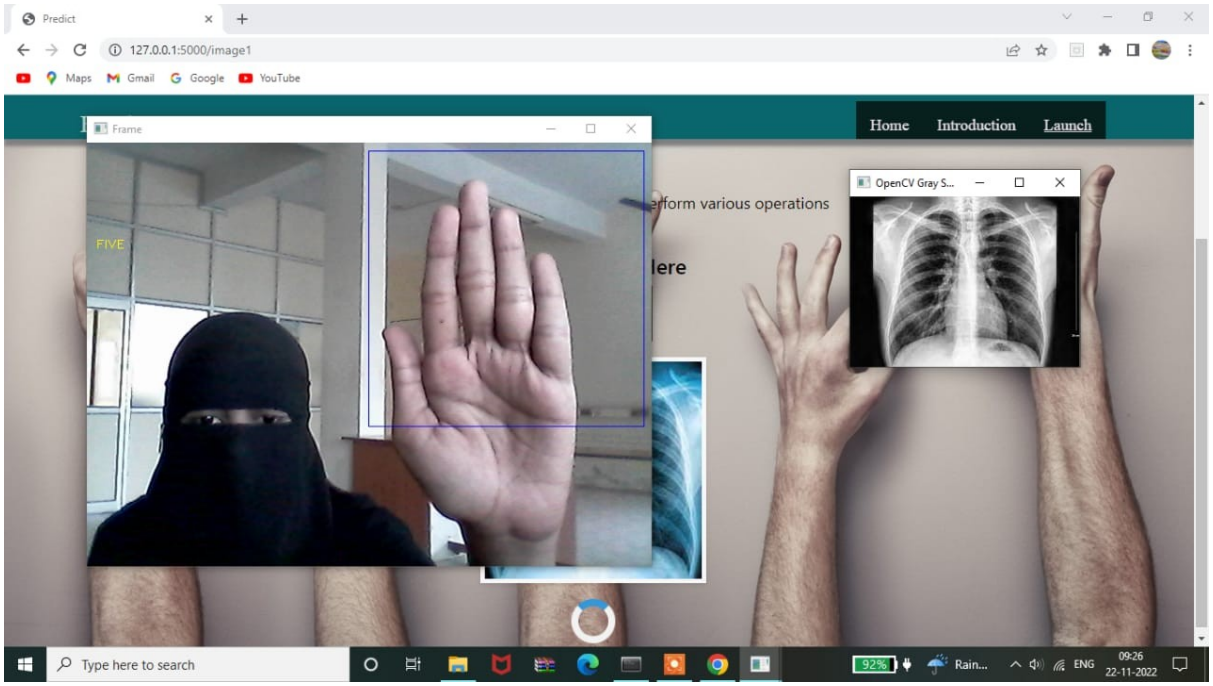
3.2 Image Upload:


Predict
127.0.0.1:5000/image1
Maps
Gmail
Google
YouTube
Hand Gesture System
Home
Introduction
Launch
Provide an image for which you want to perform various operations
Upload Image Here
Choose...
Predict!
Type here to search
89%
Rain coming
ENG
09:21
22-11-2022

3.3. Predict:


Frame
ZERO
Rectangle
Home
Introduction
Launch
esture Recognition
ich you want to perform various operations
Upload Image Here
Choose...
Type here to search
85%
Rain...
ENG
10:42
22-11-2022





4. Actions:

0 - Rectangle

1 - Fixed Re-sizing(200,200)

2 - OpenCV Rotation

3 - Blurred

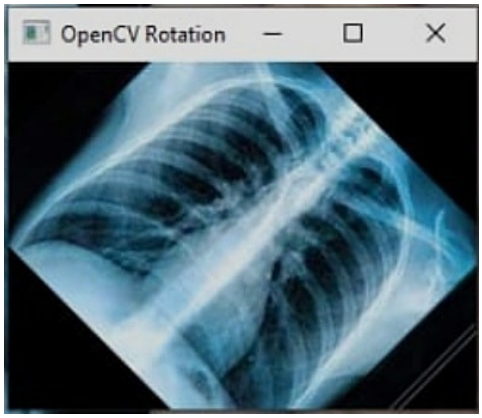
4 - Fixed Re-sizing(400,400)

5 - OpenCV Grey Scale

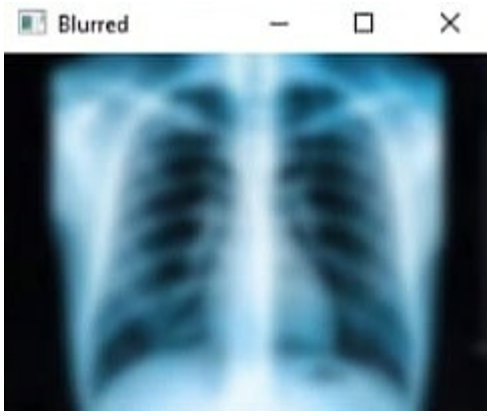
0 - Rectangle

1 - Fixed Re-sizing(200,200)

2 - OpenCV Rotation



3 - Blurred



4 - Fixed Re-sizing(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]

● Distance Control.

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 etc all 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 normalize 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 unconstrained 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 pre-made dataset has 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:

```
<html>
<script>



</script>



<style>
.header {
position: relative;
top:0;
margin:0px;
z-index: 1;
left: 0px;
right: 0px;
position: fixed;
background-color:rgb(10, 102, 109) ;
color: rgb(181, 228, 236);
box-shadow: 0px 8px 4px rgb(10, 102, 109);
```

```
overflow: hidden;
padding-left:20px;
font-family: 'Times New Roman';
font-size: 2vw;
width: 100%;
height:8%;
text-align: center;
}
.topnav {
  overflow: hidden;
  background-color: #84d39e;
}

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

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

body {

  background-color: rgb(88, 129, 123) ;
  background-repeat: no-repeat;
  background-size:cover;
  background-position: 0px 0px;
  }
  .button {
  background-color: #091425;
  border: none;
  color: rgb(181, 228, 236);
  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: 3px solid #f1f1f1; 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: rgb(181, 228, 236);
  padding: 14px 20px;
  margin-bottom:10px;
  border: none;
  cursor: pointer;
  width: 17%;
  border-radius:4px;
  font-family:Montserrat;
}

button:hover {
  opacity: 0.8;
}

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

```
}

.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(13, 53, 68);
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
 border: 10px solid rgb(13, 53, 68);
}

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

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

/* Caption text */
.text {
  color: #9ac0c0;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}
```

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

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

```
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  width: 100%;
  height: 100%;
  overflow: hidden;
  font-family: "Arial", cursive;
  -webkit-font-smoothing: antialiased;
}
::selection {
  background: transparent;
}
/* CLOUDS */
body:before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  width: 0;
  height: 0;
  margin: auto;
  border-radius: 100%;
  background: transparent;
  display: block;
  box-shadow: 0 0 150px 100px rgba(255, 255, 255, 0.6),
    200px 0 200px 150px rgba(255, 255, 255, 0.6),
    -250px 0 300px 150px rgba(255, 255, 255, 0.6),
```

```
    550px 0 300px 200px rgba(255, 255, 255, 0.6),
    -550px 0 300px 200px rgba(255, 255, 255, 0.6);
}
/* JUMP */
h1 {
  cursor: default;
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  width: 100%;
  height: 100px;
  margin: 70px;
  display: block;
  text-align: center;
}

h1 span {
  position: relative;
  top: 5px;
  display: inline-block;
  font-size: 25px;
  color: #061a1f;
   text-shadow: 0 1px 0 rgb(151, 201, 197), 0 2px 0 rgb(151, 201, 197), 0 3px 0 rgb(151, 201, 197), 0 4px 0 rgb(151, 201, 197),
    0 5px 0 rgb(151, 201, 197), 0 6px 0 transparent, 0 7px 0 transparent, 0 8px 0 transparent,
    0 9px 0 transparent, 0 10px 10px rgba(58, 159, 167, 0.4);
```

```
}

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

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

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

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

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

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

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

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

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

h1 span:nth-child(11) {
  -webkit-animation-delay: 0.5s;
}

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

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

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

/* ANIMATION */
@-webkit-keyframes bounce {
  100% {
    top: -20px;
```

```
    text-shadow: 0 1px 0 #ccc, 0 2px 0 #ccc, 0 3px 0 #ccc, 0 4px 0 #ccc,
      0 5px 0 #ccc, 0 6px 0 #ccc, 0 7px 0 #ccc, 0 8px 0 #ccc, 0 9px 0 #ccc,
      0 50px 25px rgba(0, 0, 0, 0.2);
  }
}

</style>

<body>
<h1 style="color: rgb(193, 207, 207);">
  <table style="width:100%">
    <tr>
                                                                    <th><img
src="https://media.tenor.com/images/30169e4a670daf12443df7d2dd140176/
tenor.gif" alt="NOT AVAILABLE" width="300" height="300"></th>
                                    <th><iframe width="780" height="440"
src="https://www.youtube.com/embed/nD621G8u6oc?
start=3&loop=1&autoplay=1&mute=1&controls=0">
      </iframe><br></th>
      <th></th>
    </tr>
  </table>
    <br>
    <span>HAND GESTURE RECOGNITION</span>
  </br>
    <span>OF</span>
  </br>
    <span>RADIOLOGY IMAGES</span>
  </br>
```

```
   <span>THROUGH</span>
  </br>
   <span>STERILE   BROWSING</span>

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


</body>


</html>
```

2: Intro.html:

```
<html>
<script>
```

```
</script>


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

.topnav-right a {
  float: left;
```

```
  color: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

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

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

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

body {

  background-color: white;
  background-repeat: no-repeat;
  background-size:cover;
```

```
                                                    background-image:
url("https://i.pinimg.com/originals/b2/1d/c6/b21dc69346915015bc4e19bd502f4
01b.gif");
    background-size: cover;
  background-position: 0px 0px;
  }
  .button {
  background-color: #091425;
  border: none;
  color: white;
  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: 3px solid #f1f1f1; 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: white;
  padding: 14px 20px;
  margin-bottom:10px;
  border: none;
  cursor: pointer;
  width: 17%;
  border-radius:4px;
  font-family:Montserrat;
}

button:hover {
  opacity: 0.8;
}

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

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}
```

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

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

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

.home{
                margin:80px;

  width: 84%;
```

```
  height: 500px;
  padding-top:10px;
  padding-left: 30px;

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

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

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

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

.active {
  background-color: #FCAD98;
}

/* Fading animation */
.fade {
  -webkit-animation-name: fade;
```

```
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  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}
}



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

* {
  padding: 0;
  margin: 0;
}
```

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

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

span {
  display: inline-block;
  opacity: 0;
  filter: blur(4px);
}

span:nth-child(1) {
```

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

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

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

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

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

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

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

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

```
}

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

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

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

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

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

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

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

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

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

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

```
  animation: fade-in 0.8s 2.5s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(26) {
  animation: fade-in 0.8s 2.6s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(27) {
  animation: fade-in 0.8s 2.7s forwards cubic-bezier(0.11, 0, 0.5, 0);
}span:nth-child(28) {
  animation: fade-in 0.8s 2.8s forwards cubic-bezier(0.11, 0, 0.5, 0);
}
@keyframes fade-in {
  100% {
    opacity: 1;
    filter: blur(0);
  }
}

</style>

<body>
  <h1>INTRODUCTION</h1>
<h1>
   <span> Hand Gesture </span> <span> recognition system </span> <span> provides us </span > <span> an innovative,</span> <span>natural,</span> <span>  user friendly </span> <span> way of interaction </span > <span> with the computer</span>
  <span> which is more </span> <span> familiar to the </span> <span>human beings. </span > <span> In our project, </span> <span> the hand region </span> <span> is extracted from </span> <span> the background  </span > <span> by using</span>
```

 Region of intrest.   Then,   we will be predicting 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 , blur, flip of the given image.

</h1>
<!--Brian Tracy-->

Hand Gesture System
  <div class="topnav-right"style="padding-top:0.5%;">

    Home
    <u>Introduction</u>
    Launch

</body>
</html>

3. Index.html:

<html lang="en">

<head>
  <meta charset="utf-8">

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

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

   <script  src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></
script>
  <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script   src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></
script>
       <!--   <link    href="{{    url_for('static',    filename='css\main.css')    }}"
rel="stylesheet">          -->
  <script>
    $(document).ready(function () {
      // Init
      $('.image-section').hide();
      $('.loader').hide();
      $('#result').hide();

      // Upload Preview
      function readURL(input) {
        if (input.files && input.files[0]) {
```

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

// Predict
$('#btn-predict').click(function () {
  var form_data = new FormData($('#upload-file')[0]);

  // Show loading animation
  $(this).hide();
  $('.loader').show();

  // Make prediction by calling api /predict
  $.ajax({
    type: 'POST',
    url: '/predict',
```

```
        data: form_data,
        contentType: false,
        cache: false,
        processData: false,
        async: true,
        success: function (data) {
          // Get and display the result
          $('.loader').hide();
          $('#result').fadeIn(600);
          $('#result').html(data);
          console.log('Success!');
        },
      });
    });

  });
</script>
<style>
  .bar {
    margin: 0px;
    padding: 20px;
    background-color: black;
    opacity: 0.6;
    color: black;
    font-family: 'Roboto', sans-serif;
    font-style: italic;
    border-radius: 20px;
    font-size: 25px;
  }
```

```
a {
  color: black;
  float: right;
  text-decoration: none;
  font-style: normal;
  padding-right: 20px;
}

a:hover {
  background-color: black;
  color: black;
  font-size: 30px;
  padding-left: 10px;
}

div1 {
  text-align: center;
  width: 650spx;
  height: 800px;
  padding: 190px;
  margin: 10px;
  position: absolute;
}



body {
```

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

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

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

```
.topnav-right a {
  float: left;
  color: black;
  text-align: center;
  padding: 14px 16px;
  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: 17%;
  border-radius: 4px;
  font-family: Montserrat;
}

button:hover {
  opacity: 0.8;
}

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

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}
```

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

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

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

  .cancelbtn {
    width: 100%;
  }
}

.home {
  margin: 80px;
```

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

}

.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
  }
}

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

.img-preview>div {
  width: 100%;
  height: 100%;
  background-size: cover;
```

```
  background-repeat: no-repeat;
  background-position: center;
}

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

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

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

.loader {
  border: 8px solid #f3f3f3;
  /* Light grey */
  border-top: 8px solid #3498db;
  /* Blue */
  border-radius: 50%;
```

```
      width: 50px;
      height: 50px;
      animation: spin 1s linear infinite;
    }

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

      100% {
        transform: rotate(360deg);
      }
    }
  </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 class="active" href="{{ url_for('home')}}">Home</a>
      <a class="active" href="{{ url_for('intro')}}">Introduction</a>
      <a class="active" href="{{ url_for('image1')}}"><u>Launch</u></a>
    </div>
  </div>
```

```
<br>

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

13.2 GitHub & Project Demo Link:

Project GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-17910-1659677050

Demo Link:

https://youtu.be/w8UJUTiPUUo