Gesture Based Tool for Sterile Browsing of Radiology Images

Team ID : PNT2022TMID15393

Submitted by :

S T SHARAN - 111519104134

SANJAY KUMAR S- 111519104124

SANJITH M- 111519104125

SANTHOSH K - 111519104126

CONTENTS

1. INTRODUCTION

1.1 Overview

In this project we use gestures to browse radiology images. Gestures refer to non-verbal form of communication.

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 common method of human—computer interaction. However, the use of computer keyboards and mouse by doctors and nurses in intensive care units (ICUs) is a common method for spreading infections.

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

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

1.2 PURPOSE

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

2. LITERATURE SURVEY

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

The hand gesture control system “*Gestix*” developed by the authors helped the doctor to remain in place during the entire operation, without any need to move to the main control wall since all the commands were performed using hand gestures.The sterile gesture interface consists of a Canon VC-C4 camera, whose pan/tilt/zoom can be initially set using an infrared (IR) remote. This camera is placed just over a large flat screen monitor .
Additionally, an Intel Pentium IV, (600MHz, OS: Windows XP) with a Matrox Standard II video-capturing device is used.

The “*Gibson*” image browser is a 3D visualization medical tool that enables examination of images, such as: MRIs, CT scans and X-rays. The images are arranged over a multiple layer 3D cylinder. The image of interest is found through rotating the cylinder in the four cardinal directions. To interface the gesture recognition routines with the “*Gibson*” system, information such as the centroid of the hand, its size, and orientation are used to enable screen operations in the “*Gibson*” graphical user interface.

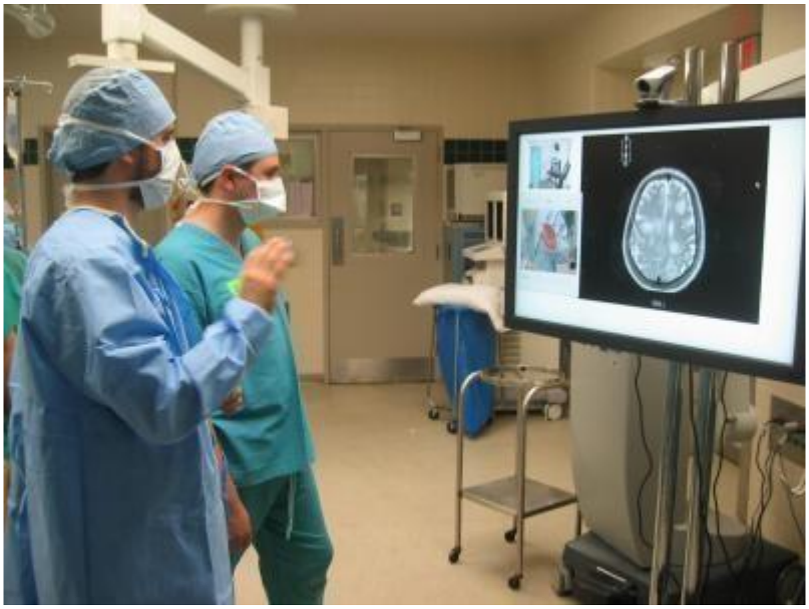
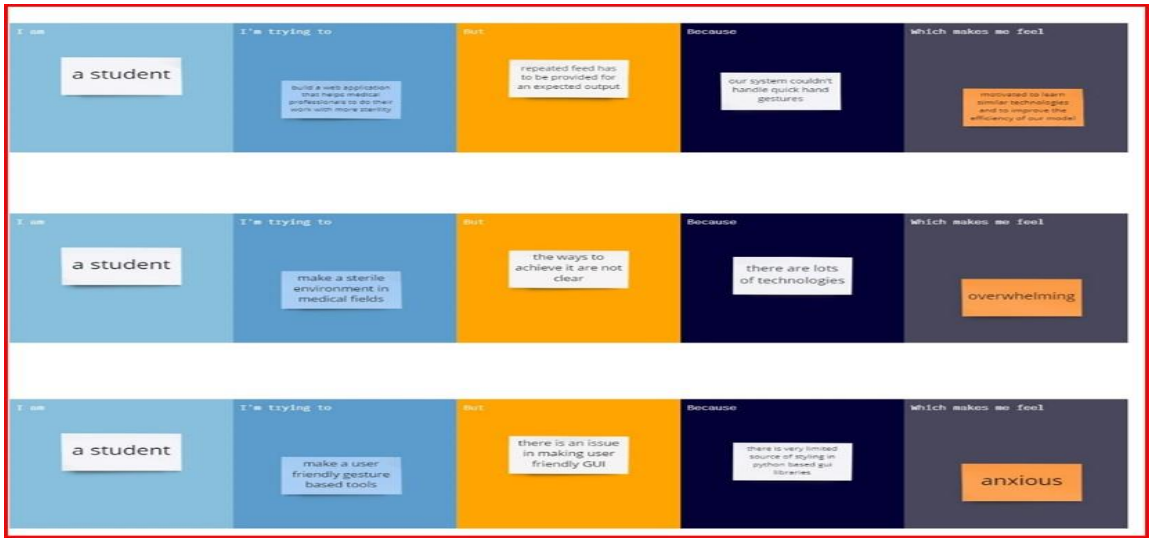


Fig 2. Radiology image browsing using hand gesture in hospital

2.2 Problem Statement Definition

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you’ll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.



I am	I'm trying to	But	Because	Which makes me feel
a student	build a web application that helps medical professionals to do their work with more sterility	repeated feed has to be provided for an expected output	our system couldn't handle quick hand gestures	motivated to learn similar technologies and to improve the efficiency of our model
a student	make a sterile environment in medical fields	the ways to achieve it are not clear	there are lots of technologies	overwhelming
a student	make a user friendly gesture based tools	there is an issue in making user friendly GUI	there is very limited source of styling in python based gui libraries	anxious

2.3 REFERENCE

1. Qing Chen Nicolas, D. Georganas, and Emil M. Petriu "Hand Gesture Recognition Using Haar-Like Features And A Stochastic Context-Free Grammar" IEEE ,Vol. 57, No. 8, August 2008.

2. Anupam Agrawal, Rohit Raj and Shubha Porwal "Vision-based Multimodal HumanComputer Interaction using Hand and Head Gestures" IEEE Conference on Information and Communication Technologies ICT 2013.

3.Kenji Oka and Yoichi Sato "Real-Time Fingertip Tracking and Gesture Recognition" IEEE proceeding on Computer Graphics and Applications Nov/Dec 2002.

4. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in International Conference on Machine Learning, 2015, pp. 448–456.

5.Juan Wachs, Helman Stern, Yael Edan, Michael Gillam, Jon Handler, Craig Feied, Mark Smith

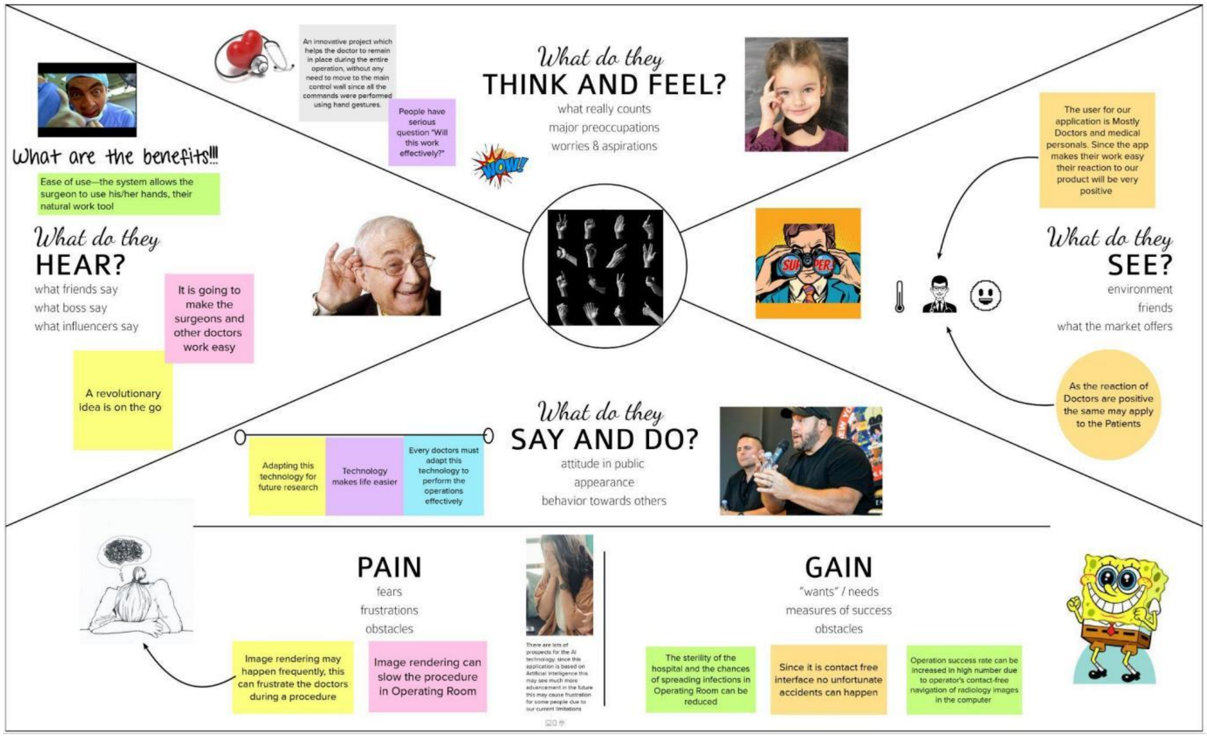
6.Professor. Juan P. Wachs,

7.Professor. Benjamin Fritsch

3 IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

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



An innovative project which helps the doctor to remain in place during the entire operation, without any need to move to the main control wall since all the commands were performed using hand gestures.

People have serious question "Will this work effectively?"

What do they THINK AND FEEL?

what really counts

major preoccupations

worries & aspirations

What are the benefits!!!

Ease of use—the system allows the surgeon to use his/her hands, their natural work tool

What do they HEAR?

what friends say

what boss say

what influencers say

It is going to make the surgeons and other doctors work easy

A revolutionary idea is on the go

The user for our application is Mostly Doctors and medical personals. Since the app makes their work easy their reaction to our product will be very positive

What do they SEE?

environment

friends

what the market offers

As the reaction of Doctors are positive the same may apply to the Patients

What do they SAY AND DO?

attitude in public

appearance

behavior towards others

Adapting this technology for future research

Technology makes life easier

Every doctors must adapt this technology to perform the operations effectively

PAIN

fears

frustrations

obstacles

Image rendering may happen frequently, this can frustrate the doctors during a procedure

Image rendering can slow the procedure in Operating Room

There are lots of prospects for the AI technology. since this application is based on Artificial intelligence this may see much more advancement in the future this may cause frustration for some people due to our current limitations

GAIN

"wants" / needs

measures of success

obstacles

The sterility of the hospital and the chances of spreading infections in Operating Room can be reduced

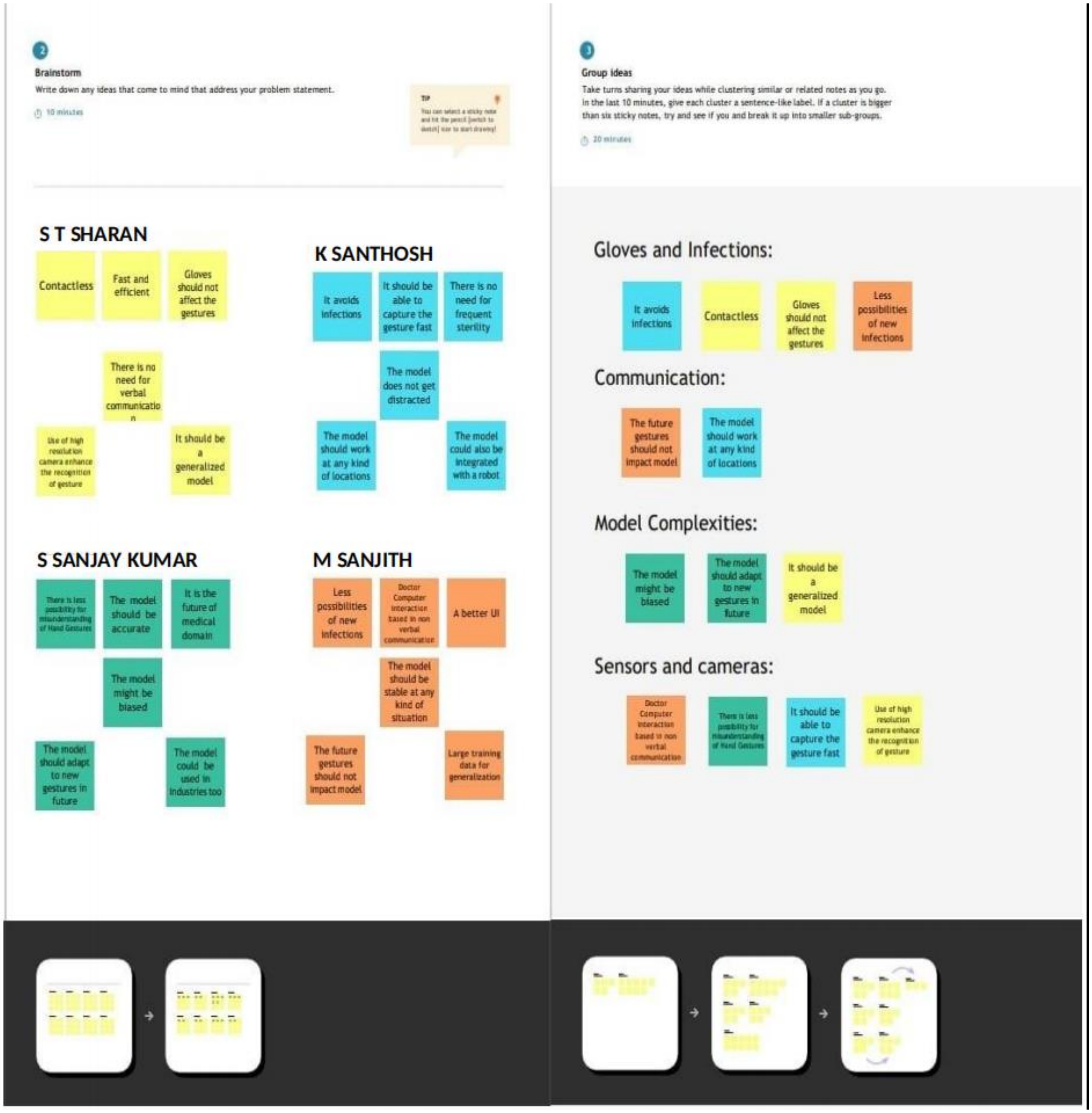
Since it is contact free interface no unfortunate accidents can happen

Operation success rate can be increased in high number due to operator's contact-free navigation of radiology images in the computer

3.2 IDEATION AND BRAINSTROM

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.

Brainstorm, Idea Listing and Grouping



2

Brainstorm

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

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

S T SHARAN

- Contactless
- Fast and efficient
- Gloves should not affect the gestures
- There is no need for verbal communicatio n
- Use of high resolution camera enhance the recognition of gesture
- It should be a generalized model

K SANTHOSH

- It avoids infections
- It should be able to capture the gesture fast
- There is no need for frequent sterility
- The model does not get distracted
- The model should work at any kind of locations
- The model could also be integrated with a robot

S SANJAY KUMAR

- There is less possibility for misunderstanding of Hand Gestures
- The model should be accurate
- It is the future of medical domain
- The model might be biased
- The model should adapt to new gestures in future
- The model could be used in Industries too

M SANJITH

- Less possibilities of new infections
- Doctor Computer interaction based in non verbal communication
- A better UI
- The model should be stable at any kind of situation
- The future gestures should not impact model
- Large training data for generalization

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Gloves and Infections:

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

Communication:

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

Model Complexities:

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

Sensors and cameras:

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

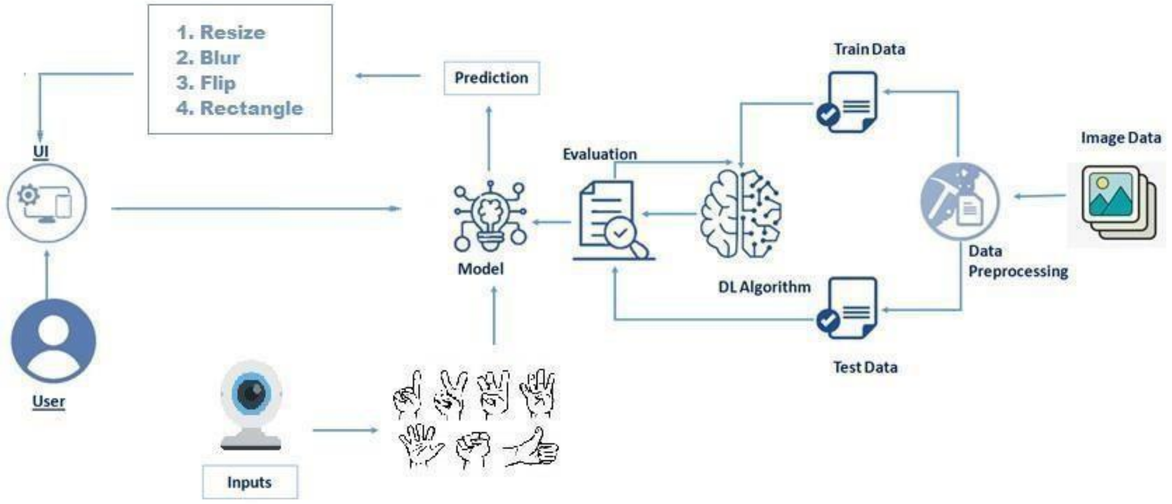
3.3 PROBLEM SOLUTION FIT

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Problem Statement	I am	I'm trying to	But	Because	Which makes me feel
PS-1	a student	build a web application that helps medical professionals to do their work with more sterility	repeated feed has to be provided for an expected output	our system couldn't handle quick hand gestures	motivated to learn similar technologies and to improve the efficiency of our model
PS-2	a student	make a sterile environment in medical fields	the ways to achieve it are not clear	there are lots of technologies	overwhelming
PS-3	a student	make a user friendly gesture based tools	there is an issue in making user friendly GUI	there is very limited source of styling in python based GUI libraries	anxious

4. REQUIREMENT ANALYSIS

4.1 REQUIREMENT ANALYSIS



4.2 NON FUNCTIONAL ANALYSIS

We found that many hospitals rely on mouse and keyboard to browse the images that are obtained during different surgeries, scans, etc. This can contaminate the environment withvarious infections thus compromising the sterility.

Various technologies have been developed to overcome this issue and one such technology was called ‘Gestix’.

This hand gesture system for MRI manipulation in an EMR image database called “*Gestix*” was tested during a brain biopsy surgery. This system is a real-time hand-tracking recognition technique based on color and motion fusion.

In an in vivo experiment, this type of interface prevented the surgeon's focus shift and change of location while achieving rapid intuitive interaction with an EMR image database. In addition to allowing sterile interaction with EMRs, the “*Gestix*” hand gesture interface provides:

1. ease of use—the system allows the surgeon to use his/her hands, their natural work tool;

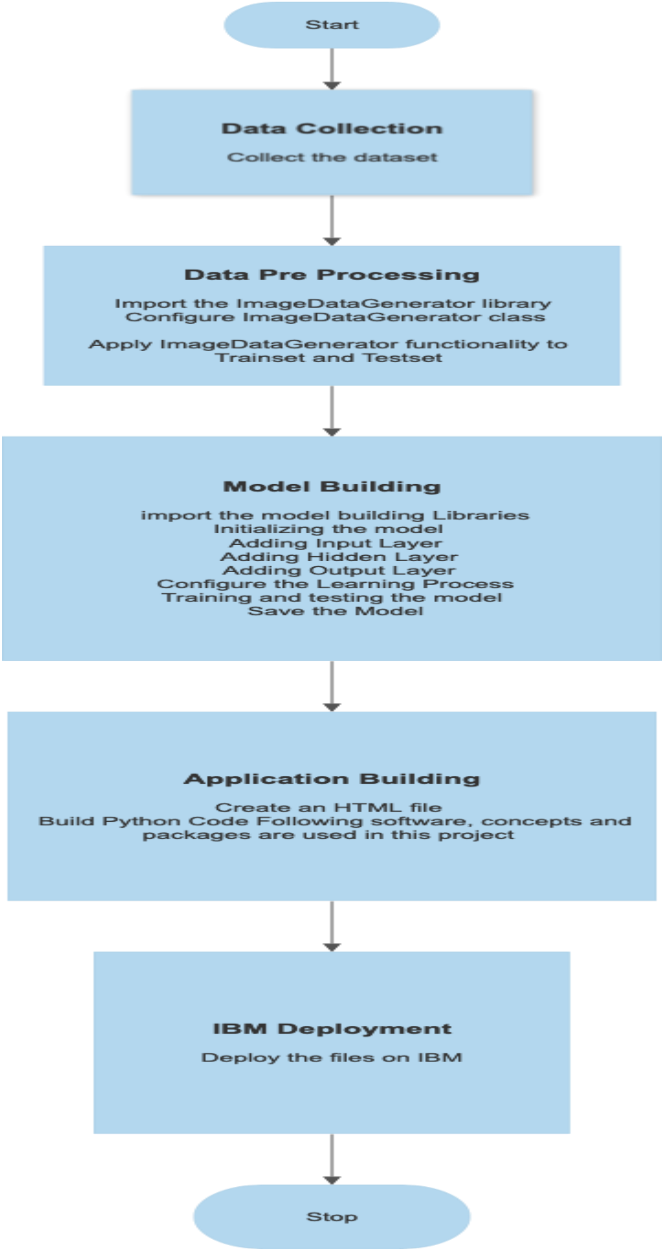
2. rapid reaction—nonverbal instructions by hand gesture commands are intuitive and fast

3. 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.

4. distance control—the hand gestures can be performed up to 5 meters from the camera and still be recognized accurately.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM


Start
Data Collection
Collect the dataset
Data Pre Processing
Import the ImageDataGenerator library
Configure ImageDataGenerator class
Apply ImageDataGenerator functionality to Trainset and Testset
Model Building
import the model building Libraries
Initializing the model
Adding Input Layer
Adding Hidden Layer
Adding Output Layer
Configure the Learning Process
Training and testing the model
Save the Model
Application Building
Create an HTML file
Build Python Code Following software, concepts and packages are used in this project
IBM Deployment
Deploy the files on IBM
Stop

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

User interacts with the UI (User Interface) to upload the image as input.

- Depending on the different gesture inputs different operations are applied to the input image.

- Once model analyses the gesture, the prediction with operation applied on image is showcased on theUI. To accomplish this, we have to complete all the activities and tasks listed below:

- Data Collection.
  - Collect the dataset or Create the dataset

- Data Pre processing
  - Import the ImageDataGenerator library
  - Configure ImageDataGenerator class
  - Apply ImageDataGenerator functionality to Trainset and Testset

- Model Building
  - Import the model building Libraries
  - Initializing the model
  - Adding Input Layer
  - Adding Hidden Layer
  - Adding Output Layer
  - Configure the Learning Process
  - Training and testing the model
  - Save the Model

- Application Building
  - Create an HTML file
  - Build Python Code Following software, concepts and packages are used in this project

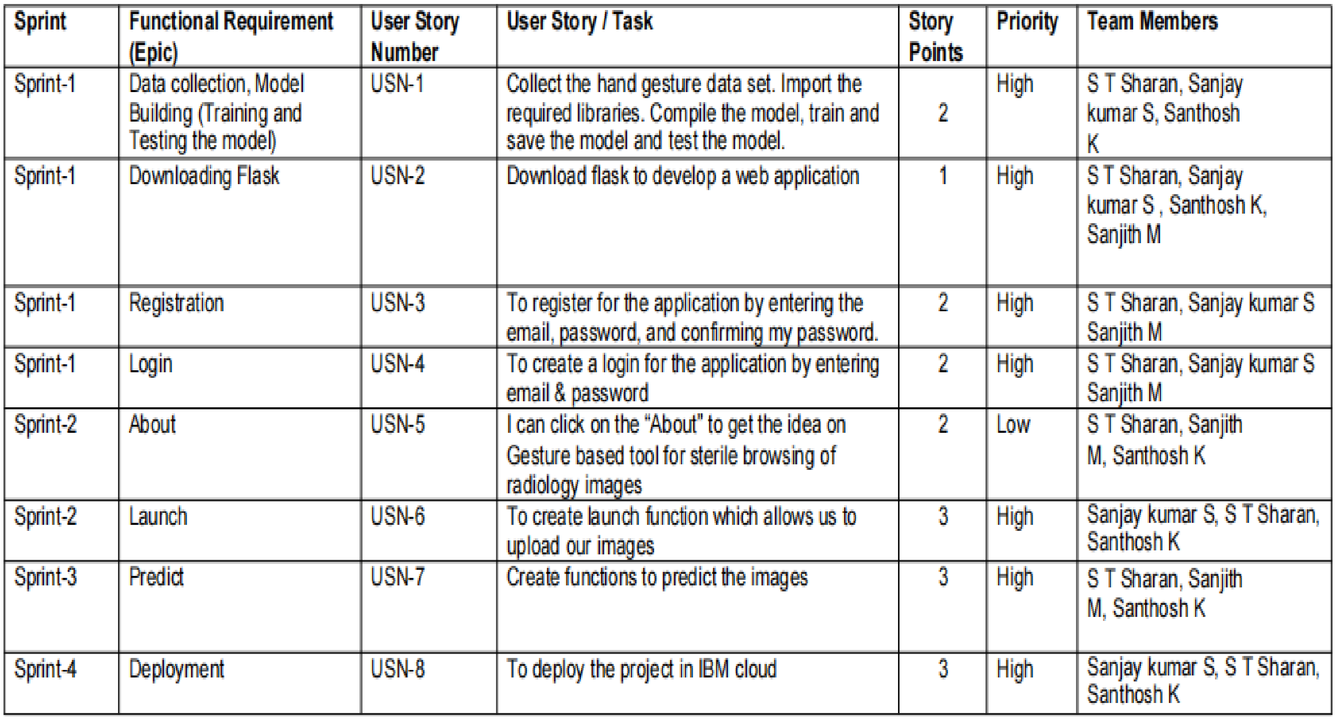
- Anaconda navigator

- Python packages:
  - open anaconda prompt as administrator
  - Type “pip install TensorFlow” (make sure you are working on python
  - 64bit)
  - Type “pip install opencv-python”
  - Type “pip install flask”

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

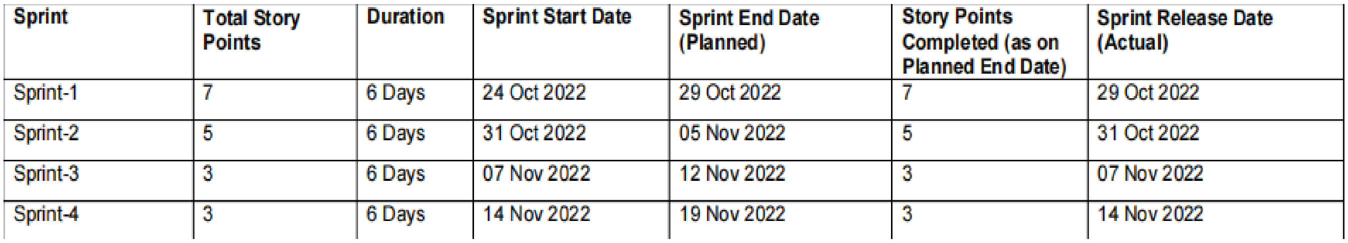
Product Backlog, Sprint Schedule, and Estimation



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data collection, Model Building (Training and Testing the model)	USN-1	Collect the hand gesture data set. Import the required libraries. Compile the model, train and save the model and test the model.	2	High	S T Sharan, Sanjay kumar S, Santhosh K
Sprint-1	Downloading Flask	USN-2	Download flask to develop a web application	1	High	S T Sharan, Sanjay kumar S , Santhosh K, Sanjith M
Sprint-1	Registration	USN-3	To register for the application by entering the email, password, and confirming my password.	2	High	S T Sharan, Sanjay kumar S Sanjith M
Sprint-1	Login	USN-4	To create a login for the application by entering email & password	2	High	S T Sharan, Sanjay kumar S Sanjith M
Sprint-2	About	USN-5	I can click on the "About" to get the idea on Gesture based tool for sterile browsing of radiology images	2	Low	S T Sharan, Sanjith M, Santhosh K
Sprint-2	Launch	USN-6	To create launch function which allows us to upload our images	3	High	Sanjay kumar S, S T Sharan, Santhosh K
Sprint-3	Predict	USN-7	Create functions to predict the images	3	High	S T Sharan, Sanjith M, Santhosh K
Sprint-4	Deployment	USN-8	To deploy the project in IBM cloud	3	High	Sanjay kumar S, S T Sharan, Santhosh K

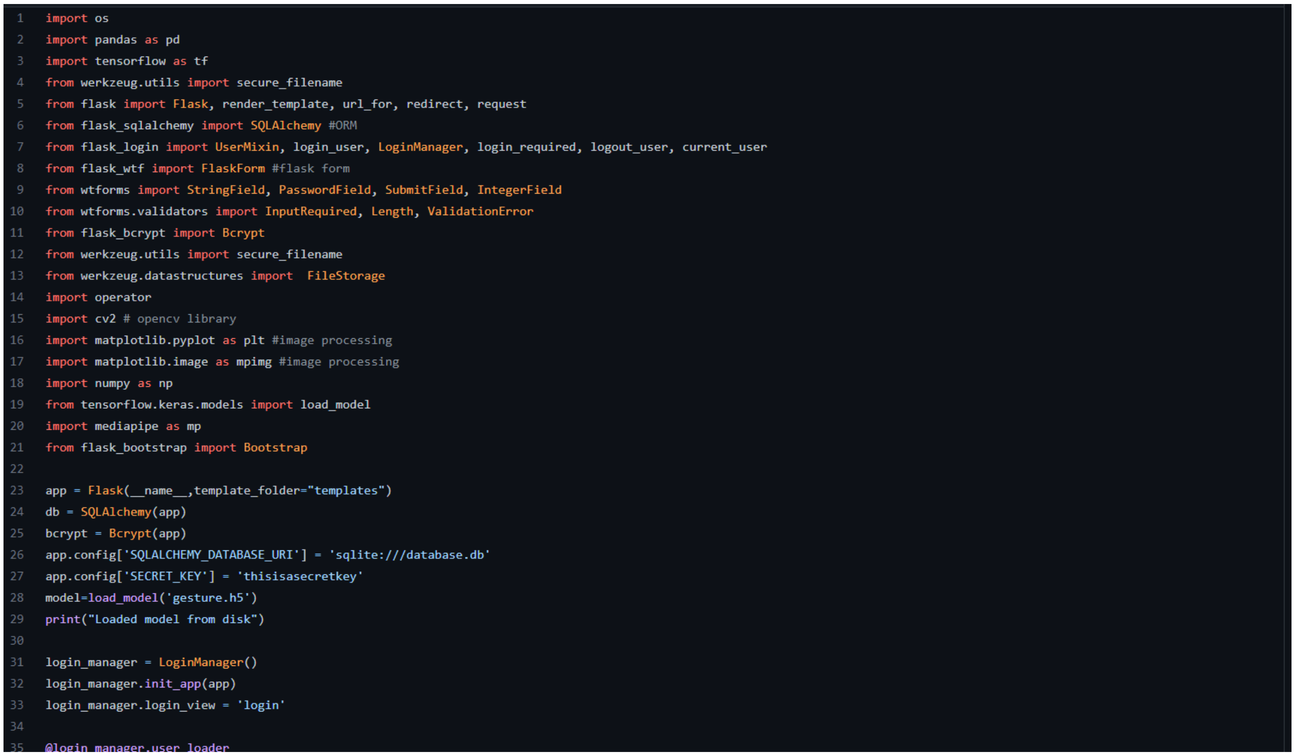
6.2 SPRINT DELIVERY SCHEDULE

Project Tracker, Velocity & Burndown Chart:

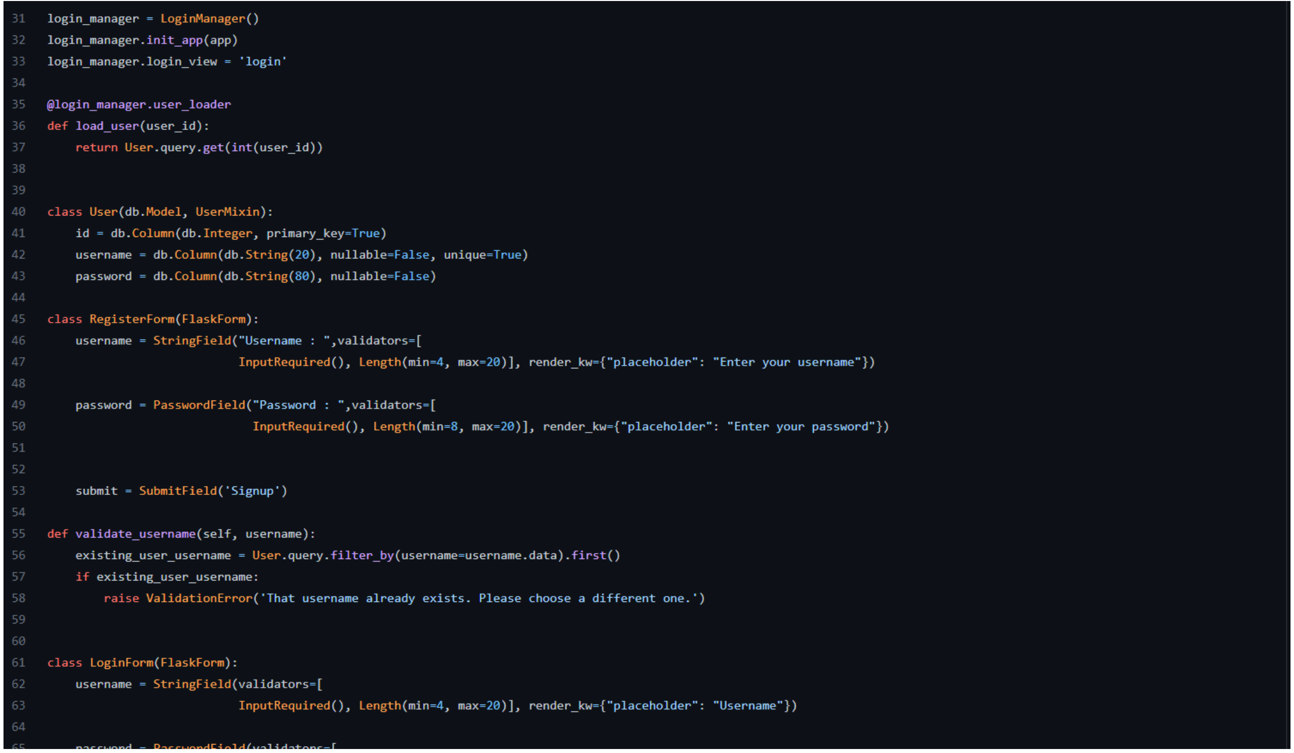


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	7	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022
Sprint-2	5	6 Days	31 Oct 2022	05 Nov 2022	5	31 Oct 2022
Sprint-3	3	6 Days	07 Nov 2022	12 Nov 2022	3	07 Nov 2022
Sprint-4	3	6 Days	14 Nov 2022	19 Nov 2022	3	14 Nov 2022

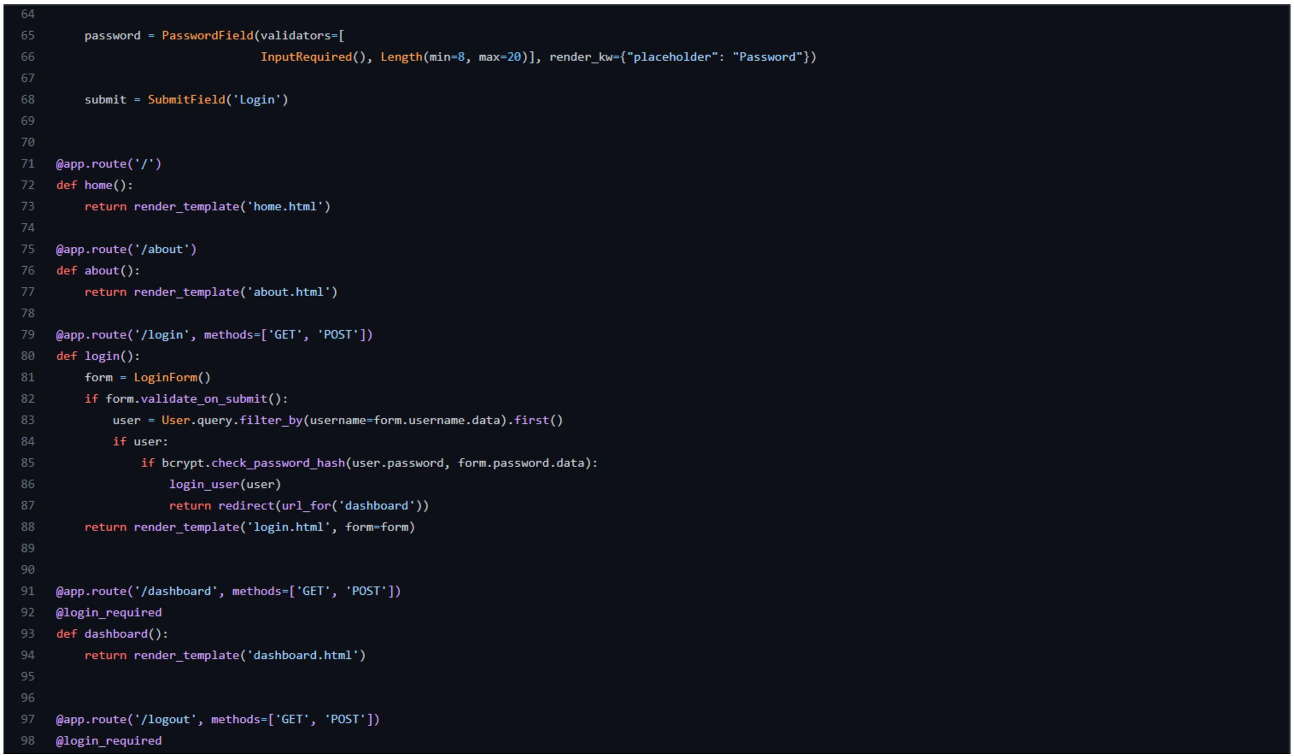
7. CODING AND SOLUTION



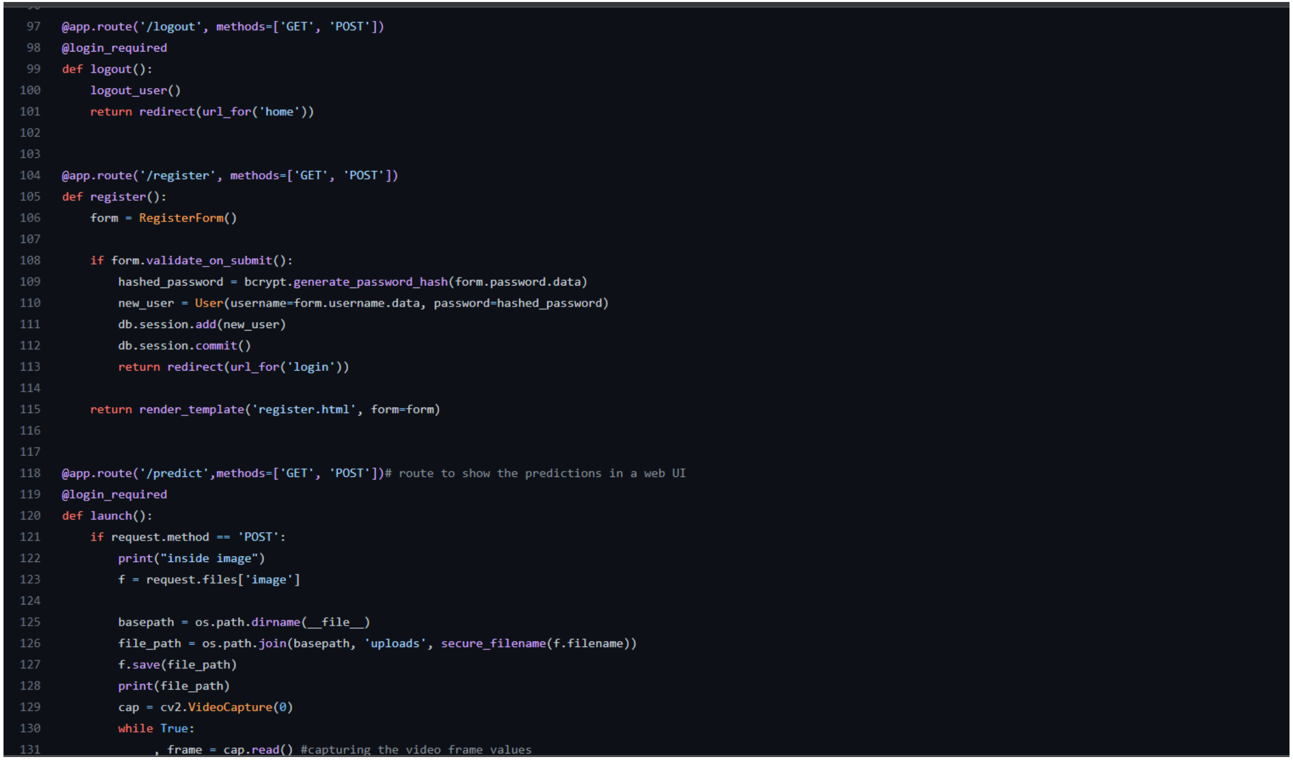
```python
import os
import pandas as pd
import tensorflow as tf
from werkzeug.utils import secure_filename
from flask import Flask, render_template, url_for, redirect, request
from flask_sqlalchemy import SQLAlchemy #ORM
from flask_login import UserMixin, login_user, LoginManager, login_required, logout_user, current_user
from flask_wtf import FlaskForm #flask form
from wtforms import StringField, PasswordField, SubmitField, IntegerField
from wtforms.validators import InputRequired, Length, ValidationError
from flask_bcrypt import Bcrypt
from werkzeug.utils import secure_filename
from werkzeug.datastructures import  FileStorage
import operator
import cv2 # opencv library
import matplotlib.pyplot as plt #image processing
import matplotlib.image as mpimg #image processing
import numpy as np
from tensorflow.keras.models import load_model
import mediapipe as mp
from flask_bootstrap import Bootstrap

app = Flask(__name__,template_folder="templates")
db = SQLAlchemy(app)
bcrypt = Bcrypt(app)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'
app.config['SECRET_KEY'] = 'thisisasecretkey'
model=load_model('gesture.h5')
print("Loaded model from disk")

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'

@login_manager.user_loader
```



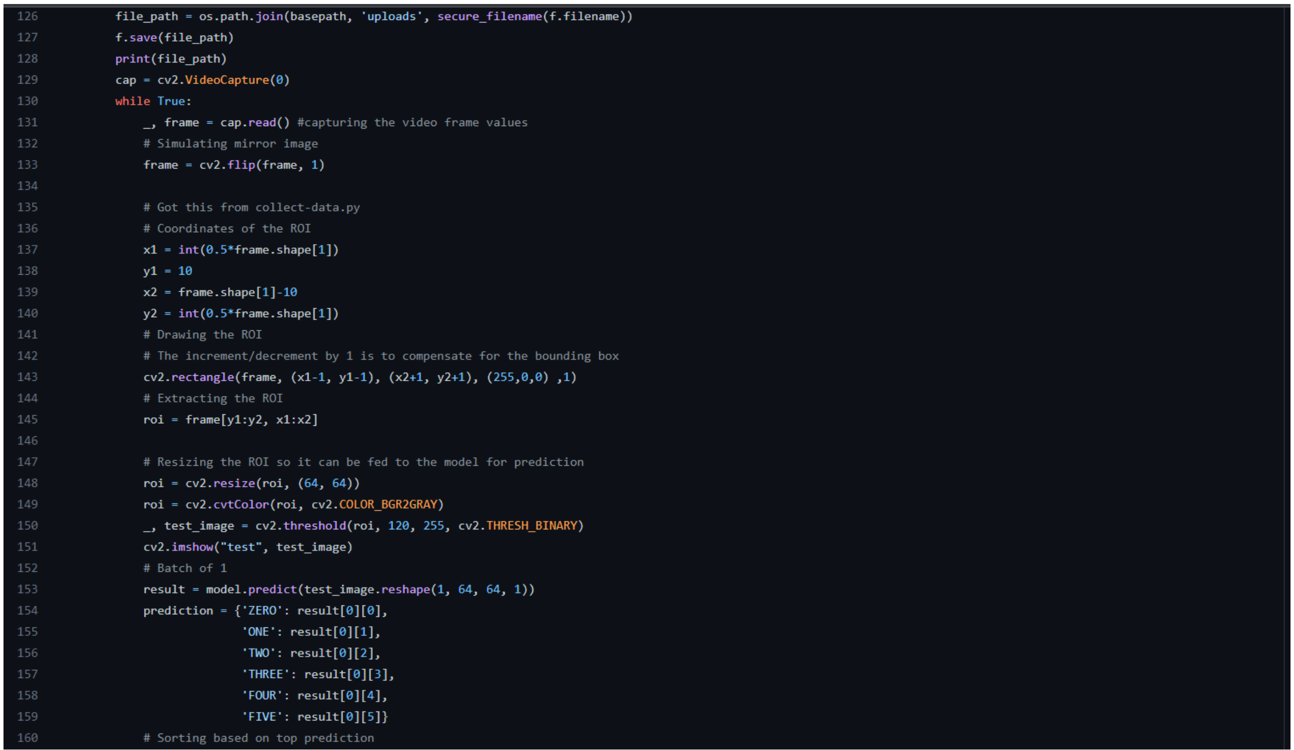
```python
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'

@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))


class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(20), nullable=False, unique=True)
    password = db.Column(db.String(80), nullable=False)

class RegisterForm(FlaskForm):
    username = StringField("Username : ",validators=[
                           InputRequired(), Length(min=4, max=20)], render_kw={"placeholder": "Enter your username"})

    password = PasswordField("Password : ",validators=[
                             InputRequired(), Length(min=8, max=20)], render_kw={"placeholder": "Enter your password"})


    submit = SubmitField('Signup')

def validate_username(self, username):
    existing_user_username = User.query.filter_by(username=username.data).first()
    if existing_user_username:
        raise ValidationError('That username already exists. Please choose a different one.')


class LoginForm(FlaskForm):
    username = StringField(validators=[
                           InputRequired(), Length(min=4, max=20)], render_kw={"placeholder": "Username"})

    password = PasswordField(validators=[
```



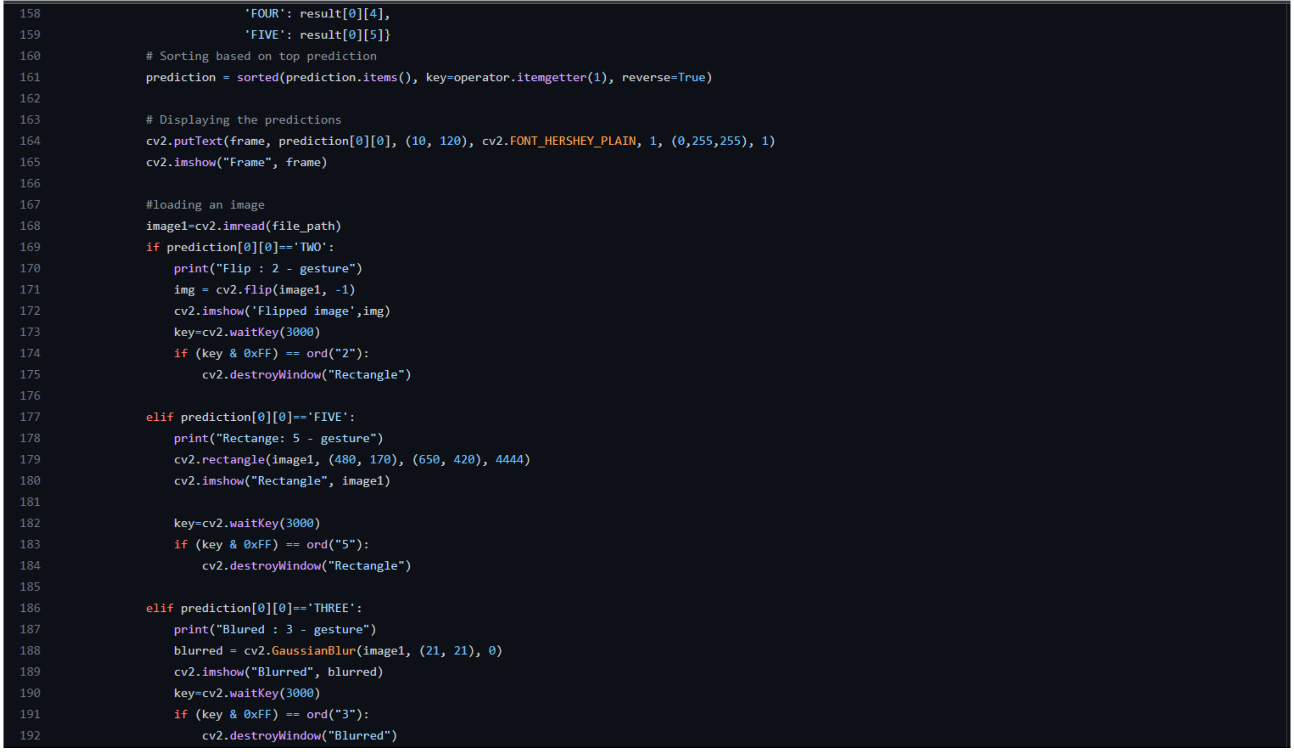
```python

    password = PasswordField(validators=[
                             InputRequired(), Length(min=8, max=20)], render_kw={"placeholder": "Password"})

    submit = SubmitField('Login')


@app.route('/')
def home():
    return render_template('home.html')

@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user:
            if bcrypt.check_password_hash(user.password, form.password.data):
                login_user(user)
                return redirect(url_for('dashboard'))
    return render_template('login.html', form=form)


@app.route('/dashboard', methods=['GET', 'POST'])
@login_required
def dashboard():
    return render_template('dashboard.html')


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



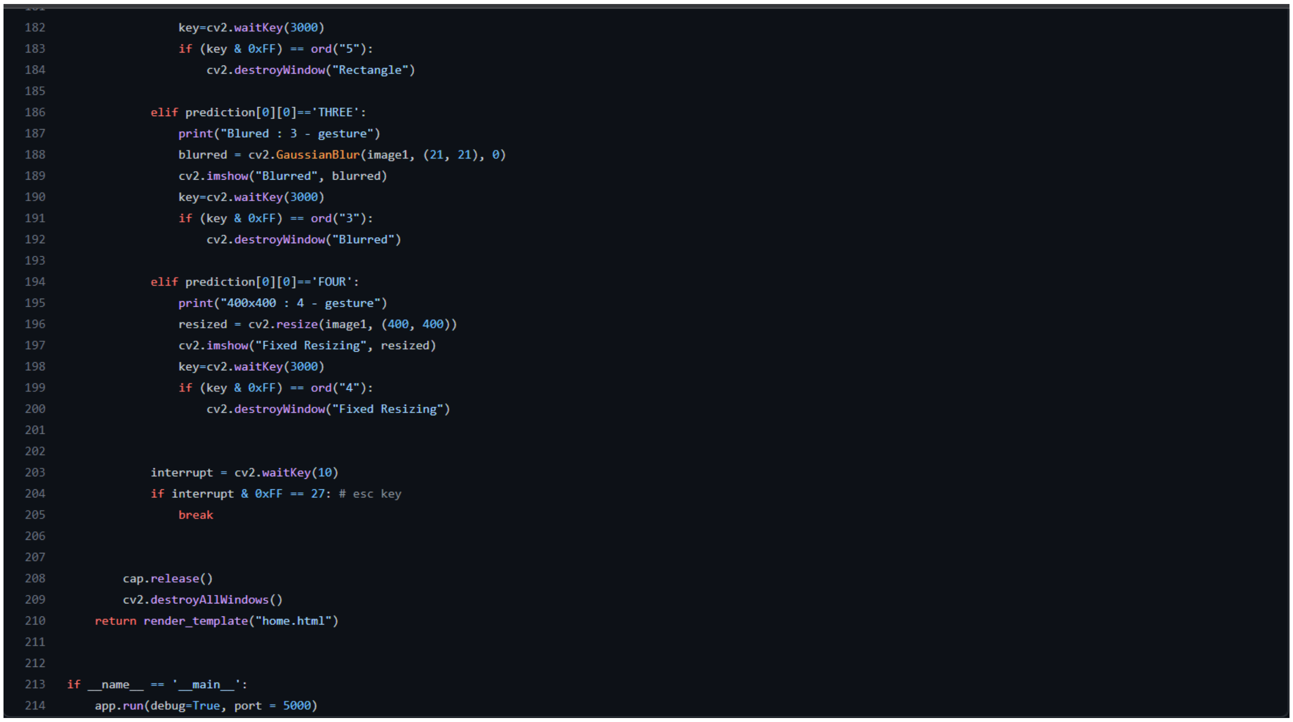
```python
@app.route('/logout', methods=['GET', 'POST'])
@login_required
def logout():
    logout_user()
    return redirect(url_for('home'))


@app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegisterForm()

    if form.validate_on_submit():
        hashed_password = bcrypt.generate_password_hash(form.password.data)
        new_user = User(username=form.username.data, password=hashed_password)
        db.session.add(new_user)
        db.session.commit()
        return redirect(url_for('login'))

    return render_template('register.html', form=form)


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



```python
        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],
                          '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
```



```python
                          '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]=='TWO':
                print("Flip : 2 - gesture")
                img = cv2.flip(image1, -1)
                cv2.imshow('Flipped image',img)
                key=cv2.waitKey(3000)
                if (key & 0xFF) == ord("2"):
                    cv2.destroyWindow("Rectangle")

            elif prediction[0][0]=='FIVE':
                print("Rectange: 5 - gesture")
                cv2.rectangle(image1, (480, 170), (650, 420), 4444)
                cv2.imshow("Rectangle", image1)

                key=cv2.waitKey(3000)
                if (key & 0xFF) == ord("5"):
                    cv2.destroyWindow("Rectangle")

            elif prediction[0][0]=='THREE':
                print("Blured : 3 - gesture")
                blurred = cv2.GaussianBlur(image1, (21, 21), 0)
                cv2.imshow("Blurred", blurred)
                key=cv2.waitKey(3000)
                if (key & 0xFF) == ord("3"):
                    cv2.destroyWindow("Blurred")
```

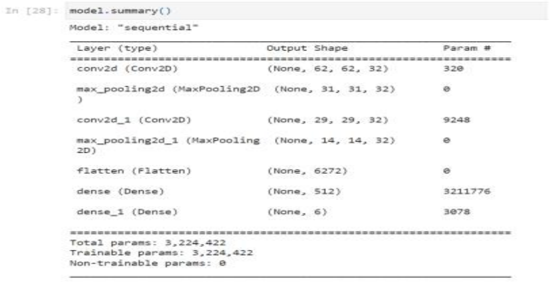
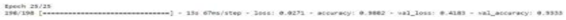


```
                key=cv2.waitKey(3000)
                if (key & 0xFF) == ord("5"):
                    cv2.destroyWindow("Rectangle")

            elif prediction[0][0]=='THREE':
                print("Blured : 3 - gesture")
                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':
                print("400x400 : 4 - gesture")
                resized = cv2.resize(image1, (400, 400))
                cv2.imshow("Fixed Resizing", resized)
                key=cv2.waitKey(3000)
                if (key & 0xFF) == ord("4"):
                    cv2.destroyWindow("Fixed Resizing")


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


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


if __name__ == '__main__':
    app.run(debug=True, port = 5000)
```

8.TESTING

User Acceptance Testing

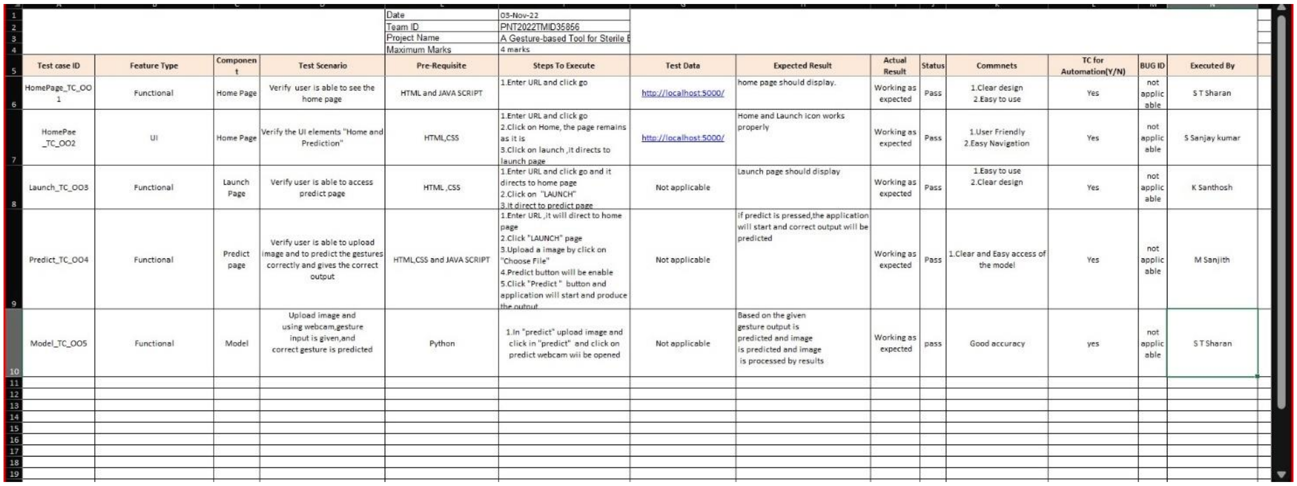
Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	
2.	Accuracy	Training Accuracy – 0.9882 Validation Accuracy – 0.9333	

```
In [28]: model.summary()
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        320

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

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

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

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 512)               3211776

 dense_1 (Dense)             (None, 6)                 3078

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

```
Epoch 25/25
198/198 [==============================] - 13s 67ms/step - loss: 0.0271 - accuracy: 0.9882 - val_loss: 0.4183 - val_accuracy: 0.9333
```

8.2 Testing Test Cases

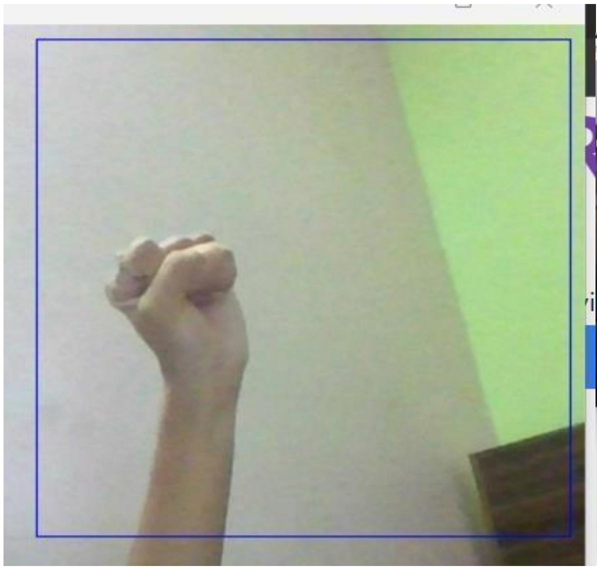


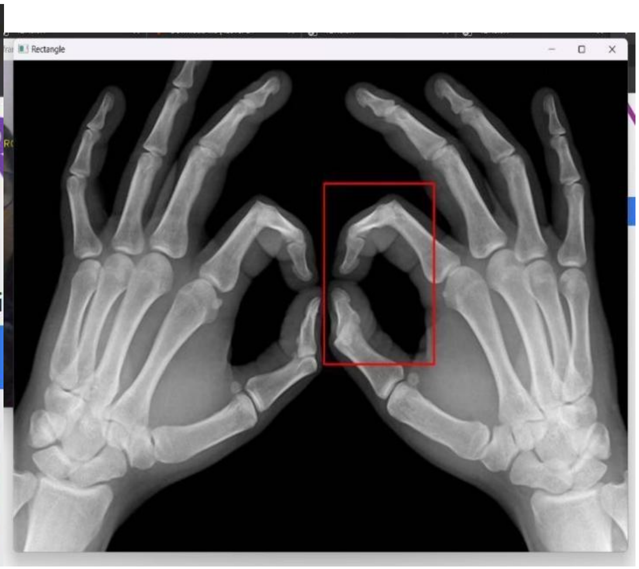
Date	03-Nov-22
Team ID	PNT2022TMID35856
Project Name	A Gesture-based Tool for Sterile E
Maximum Marks	4 marks

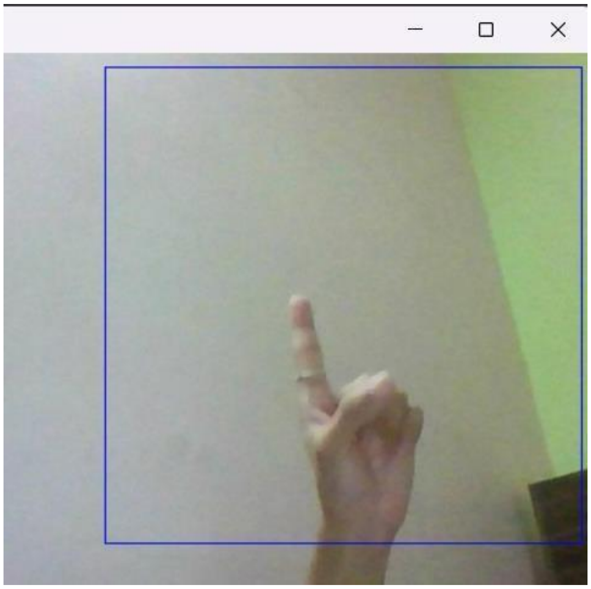
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_OO1	Functional	Home Page	Verify user is able to see the home page	HTML and JAVA SCRIPT	1.Enter URL and click go	http://localhost:5000/	home page should display.	Working as expected	Pass	1.Clear design 2.Easy to use	Yes	not applicable	S T Sharan
HomePae _TC_OO2	UI	Home Page	Verify the UI elements "Home and Prediction"	HTML,CSS	1.Enter URL and click go 2.Click on Home, the page remains as it is 3.Click on launch ,it directs to launch page	http://localhost:5000/	Home and Launch icon works properly	Working as expected	Pass	1.User Friendly 2.Easy Navigation	Yes	not applicable	S Sanjay kumar
Launch_TC_OO3	Functional	Launch Page	Verify user is able to access predict page	HTML ,CSS	1.Enter URL and click go and it directs to home page 2.Click on "LAUNCH" 3.It direct to predict page	Not applicable	Launch page should display	Working as expected	Pass	1.Easy to use 2.Clear design	Yes	not applicable	K Santhosh
Predict_TC_OO4	Functional	Predict page	Verify user is able to upload image and to predict the gestures correctly and gives the correct output	HTML,CSS and JAVA SCRIPT	1.Enter URL ,it will direct to home page 2.Click "LAUNCH" page 3.Upload a image by click on "Choose File" 4.Predict button will be enable 5.Click "Predict " button and application will start and produce the output	Not applicable	if predict is pressed,the application will start and correct output will be predicted	Working as expected	Pass	1.Clear and Easy access of the model	Yes	not applicable	M Sanjith
Model_TC_OO5	Functional	Model	Upload image and using webcam,gesture input is given,and correct gesture is predicted	Python	1.In "predict" upload image and click in "predict" and click on predict webcam wii be opened	Not applicable	Based on the given gesture output is predicted and image is predicted and image is processed by results	Working as expected	pass	Good accuracy	yes	not applicable	S T Sharan

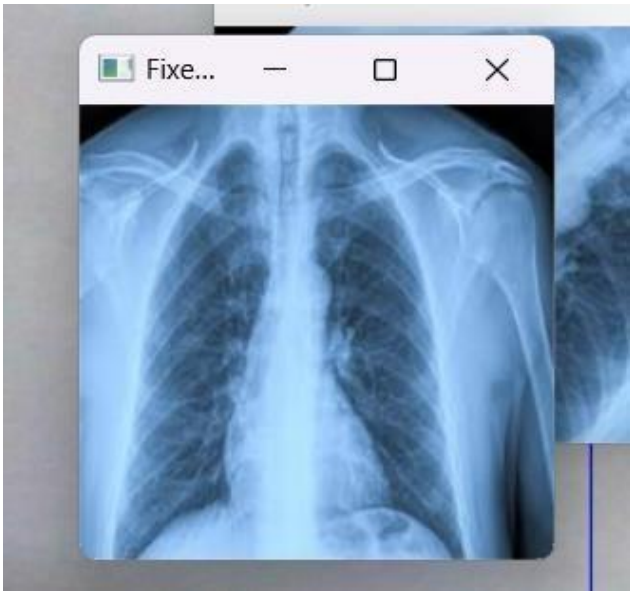
9.RESULT

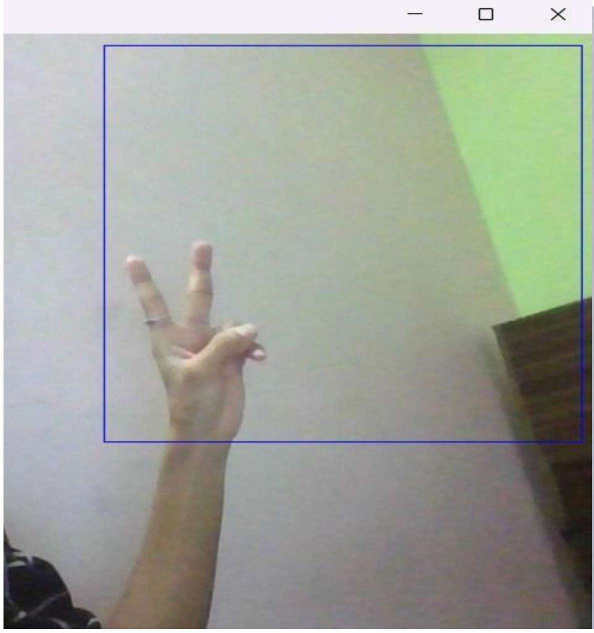
Final findings (Output) of the project along with screenshots.

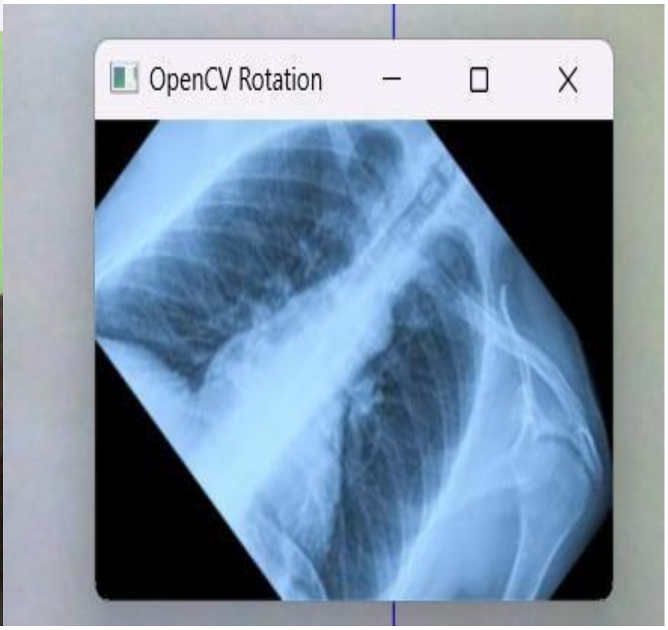


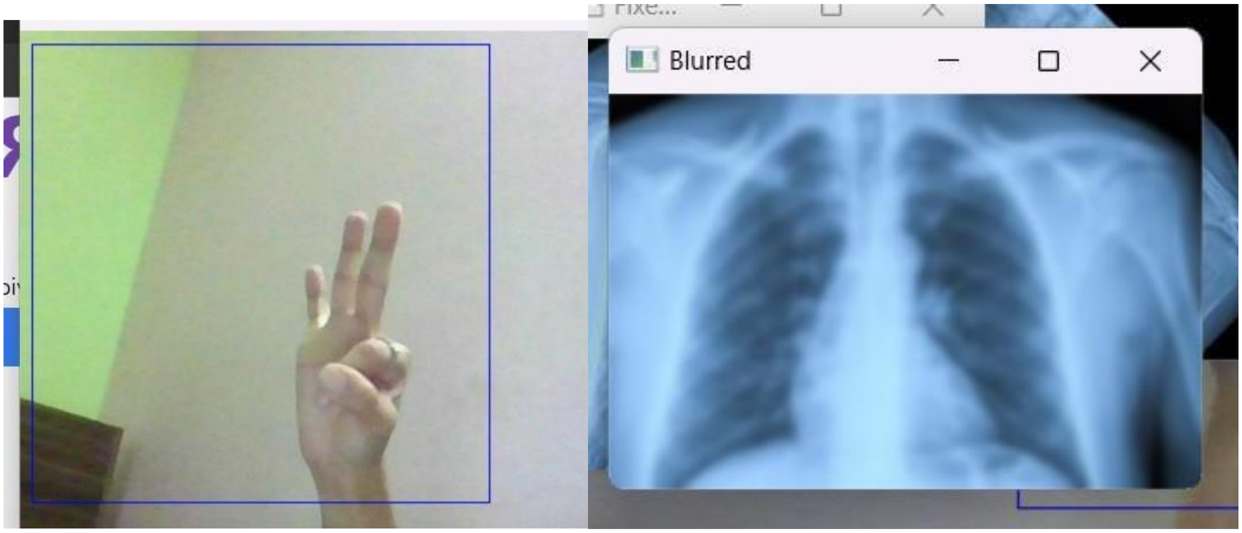





Fixe...




OpenCV Rotation



Through this project we found that we can maintain the sterility of an operation theater, etc by using hand based gesture tools to browse the images obtained.

10.ADVANTAGES & DISADVANTAGES

Advantages:

- Major advantage of this tool is that it helps to maintain the sterility of the environment.
- It is also easy to use and is quicker than the existing methods to browse images.
- It can also be performed even if the surgeon is a bit far away from the system, this helps to save time.
- The tool does not need the person using it to have an apparatus or any devices on them to use it. They can simply move their hands to browse through the images.

Disadvantages:

- The tool can be quite expensive as it requires cameras and other expensive devices to capture images and process it.

APPLICATIONS

- This hand based gesture tool developed can be mainly used in the medical industry to browse images without compromising the sterility.
- However it can also be used in different industries while presenting certain ideas, during

meetings, and can be used by teachers while teaching.

11.CONCLUSION

In this project we developed a tool which recognises hand gestures and enables doctors to browse through radiology images using these gestures. This enables doctors and surgeons to maintain the sterility as they would not have to touch any mouse or keyboard to go through the images.

This tool is also easy to use and is quicker than the regular method of using mouse/keyboard.

It can be used regardless of the users location since they don't have to be in contact with any device.

It also does not require the user to have any device on them to use it.

Further this technology can be extended to other industries like it can be used by presenters, by teachers for show images in the classroom, etc.

12.FUTURE SCOPE

- The tool can be made quicker by increasing the recognition speed.
- More number of gestures can be added thereby increasing this tool's functionality and useability for different purposes.
- Tracking of both hands can be added to increase the set of commands. Voice commands can also be added to further increase the functionality.

13.Appendix:

13.1 Source code:

```
import os
import pandas as pd
import tensorflow as tf
from werkzeug.utils import secure_filename
from flask import Flask, render_template, url_for, redirect, request
from flask_sqlalchemy import SQLAlchemy #ORM
from flask_login import UserMixin, login_user, LoginManager, login_required, logout_user,
current_user
from flask_wtf import FlaskForm #flask form
from wtforms import StringField, PasswordField, SubmitField, IntegerField
from wtforms.validators import InputRequired, Length, ValidationError
from flask_bcrypt import Bcrypt
from werkzeug.utils import secure_filename
from werkzeug.datastructures import  FileStorage
import operator
import cv2 # opencv library
import matplotlib.pyplot as plt #image processing
import matplotlib.image as mpimg #image processing
import numpy as np
from tensorflow.keras.models import load_model
import mediapipe as mp
from flask_bootstrap import Bootstrap

app = Flask(__name__,template_folder="templates")
db = SQLAlchemy(app)
bcrypt = Bcrypt(app)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'
app.config['SECRET_KEY'] = 'thisisasecretkey'
model=load_model('gesture.h5')
print("Loaded model from disk")

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'

@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))



class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(20), nullable=False, unique=True)
    password = db.Column(db.String(80), nullable=False)

class RegisterForm(FlaskForm):
    username = StringField("Username : ",validators=[
                        InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":
"Enter your username"})

    password = PasswordField("Password : ",validators=[
                         InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Enter your password"})
```

```
    submit = SubmitField('Signup')

def validate_username(self, username):
    existing_user_username = User.query.filter_by(username=username.data).first()
    if existing_user_username:
        raise ValidationError('That username already exists. Please choose a different one.')


class LoginForm(FlaskForm):
    username = StringField(validators=[
                           InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":
"Username"})

    password = PasswordField(validators=[
                             InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Password"})

    submit = SubmitField('Login')


@app.route('/')
def home():
    return render_template('home.html')

@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user:
            if bcrypt.check_password_hash(user.password, form.password.data):
                login_user(user)
                return redirect(url_for('dashboard'))
    return render_template('login.html', form=form)
```

```
@app.route('/dashboard', methods=['GET', 'POST'])
@login_required
def dashboard():
    return render_template('dashboard.html')


@app.route('/logout', methods=['GET', 'POST'])
@login_required
def logout():
    logout_user()
    return redirect(url_for('home'))


@app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegisterForm()

    if form.validate_on_submit():
        hashed_password = bcrypt.generate_password_hash(form.password.data)
        new_user = User(username=form.username.data, password=hashed_password)
        db.session.add(new_user)
        db.session.commit()
        return redirect(url_for('login'))

    return render_template('register.html', form=form)


@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
@login_required
def launch():
    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],
                  '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]=='TWO':
    print("Flip : 2 - gesture")
    img = cv2.flip(image1, -1)
    cv2.imshow('Flipped image',img)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("2"):
        cv2.destroyWindow("Rectangle")

elif prediction[0][0]=='FIVE':
    print("Rectange: 5 - gesture")
    cv2.rectangle(image1, (480, 170), (650, 420), 4444)
    cv2.imshow("Rectangle", image1)

    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("5"):
        cv2.destroyWindow("Rectangle")

elif prediction[0][0]=='THREE':
    print("Blured : 3 - gesture")
    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':
    print("400x400 : 4 - gesture")
    resized = cv2.resize(image1, (400, 400))
    cv2.imshow("Fixed Resizing", resized)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("4"):
        cv2.destroyWindow("Fixed Resizing")
```

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


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


if __name__ == '__main__':
    app.run(debug=True, port = 5000)
```

13.2 GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-31294-1660198703

13.3 Demo link:

https://youtu.be/gQI3hhdo_R0