A GESTURE BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

TEAM MEMBERS
PADMA PRIYA T R – 962319104065 (TEAM LEAD)
THANYA S S – 962319104094 (MEMBER 1)
NARESH K – 962319104061 (MEMBER 2)
PRANAV R – 962319104068 (MEMBER 3)

1. INTRODUCTION

A gesture based automation tool for sterile browsing of radiology images helps in processing the radiology images by means of gestures and without having any physical contact with the computing devices such as keyboards, mouse etc.

1.1 Project Overview

Humans are able to recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development. In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others. In this project Gesture based Desktop automation, first the CNN 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 1 then images is resized. If 2 is predicted, image is blurred, 3 for image rotation and 4 for rectangle.

1.2 Purpose

The purpose of this project is to do the following.

- To assists surgeons while performing operations at a fast rate without any physical contact.
- Make use of hand gestures as an alternative to existing interface techniques, thus offering the major advantage of sterility.
- To highly benefit customers by performing surgeries without touching any pointing devices and also to save time.

2. LITERATURE SURVEY

A gesture-based tool for sterile browsing of radiology images (2008) by JUAN P. WACHS, PHD, HELMAN I. STERN, PHD, YAEL EDAN, PHD, MICHAEL GILLAM, MD, JON HANDLER, MD, CRAIG FEIED, MD, PHD, MARK SMITH, MD - "Gestix," a vision-based hand gesture capture and recognition system that interprets in real-time the user's gestures for navigation and Manipulation of images in an electronic medical record (EMR) database is used. Navigation and other gestures are translated to commands based on their temporal trajectories, through video capture.

Hand-gesture-based sterile interface for the operating room using contextual cues for the navigation of radiological images (2012) by Mithun George Jacob, Juan Pablo Wachs, Rebecca A Packer. Computer vision algorithms were developed to extract intention and attention cues from the surgeon's behavior and combine them with sensory data from a commodity depth camera.

REALISM: Real-Time Hand Gesture Interface for Surgeons and Medical Experts (2010) by David Louis M. Achacon Jr., Denise M. Carlos, Maryann Kaye Puyaoan Christine T. Clarin, Prospero C. Naval, Jr. The project was developed using OpenCV, a computer vision library originally developed by Intel and Cascade Classifiers, PCA and Nearest Distance Matching.

S. No	Title	Merits	Demerits

1	A gesture-based tool for sterile browsing of radiology images	Ease of use, Rapid reaction, An unencumbered interface, Distance control—the hand gestures can be performed up to 5 meters from the camera	The setup time for the whole "Gestix" system was approximately 20 minutes.
2	Hand-gesture-based sterile interface for the operating room using contextual cues for the navigation of radiological images	It uses environmental cues to determine intent allowing the user to perform gestures anywhere in the field of view of the sensor and also the framework can be extended to a large gesture vocabulary.	The tracking algorithm occasionally failed in the presence of several people in the camera field of view.
3	REALISM: Real-Time Hand Gesture Interface for Surgeons and Medical Experts	The hand detection module was able to achieve higher precision and recall in well lighted environment.	Although the system detects most of the hands present in the camera's vision, it still misclassifies some objects as hands. The performance was not that good in poorly illuminated environment

2.1 Existing problem

One of the problems in existing system is that the "Gestix" system takes approximately 20 minutes for setup. The tracking algorithm occasionally failed in the presence of several people in the camera field of view. Although the system detects most of the hands present in the camera's vision, it still misclassifies some objects as hands. The performance was not that good in poorly illuminated environment

2.2 References

- A gesture-based tool for sterile browsing of radiology images by JUAN P. WACHS, PHD, HELMAN I. STERN, PHD, YAEL EDAN, PHD, MICHAEL GILLAM, MD, JON HANDLER, MD, CRAIG FEIED, MD, PHD, MARK SMITH, MD
- Hand-gesture-based sterile interface for the operating room using contextual cues for the navigation of radiological images by Mithun George Jacob, Juan Pablo Wachs, Rebecca A Packer.
- REALISM: Real-Time Hand Gesture Interface for Surgeons and Medical Experts by David Louis M. Achacon Jr., Denise M. Carlos, Maryann Kaye Puyaoan Christine T. Clarin, Prospero C. Naval, Jr.

2.3 Problem Statement Definition

Computer information technology is increasingly penetrating into the hospital domain. 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. Even though voice control also provides sterility, the noise level in the operating room (OR) deems it problematic. Our solution for this problem is the use of hand gestures as an alternative to existing interface techniques, offering the major advantage of sterility. In our work we refer to gestures as a basic form of non-verbal communication made with the hands.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS - 1	Doctors	Perform surgery by sterile browsing of radiology images	The gestures may be misinterpreted	Camera may detect multiple persons in the OR	Frustrated
PS - 2	Nurses	Perform surgery by sterile browsing of radiology images	It may predict wrong under bad lightings	Camera hardly detects images in darkness	Anxious

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Empathy Map

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

- Will the video be captured correctly
- Is the system user-friendly ?
- Is the system bug free?
- Will the system produce correct output?
- Surgeries go wrong if the output is wrong
- Users should be benefited

What do they HEAR?
what friends say
what boss say
what influencers say

- UI should be flexible
- Check the working in different lightings
- It should create a buzz in the market

What do they SEE?
environment
friends
what the market offers

- Video Capturing
- Correctness of output
- Interaction with the system
- Quality of images produced
- Interactive User Interface

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

- What are the gestures to be added?
- Compare against existing systems
- Algorithms to be used to develop the system
- Testing should be done frequently

PAIN
fears
frustrations
obstacles

- Limited Hand gestures
- Limited to one person
- Poor performance in bad lightings
- Limited to certain distance

GAIN
"wants" / needs
measures of success
obstacles

- User-friendly
- Time Saving
- Cost effective
- Accurate Predictions

3.2 Ideation & Brainstorming

1

Define your problem statement

PROBLEM

Keyboards, mouse etc 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 is a common method for spreading infections. Even though voice control also provides sterility, the noise level in the operating room deems it problematic. To overcome this, our system makes use of hand gestures as an alternative to existing interface techniques, offering the major advantage of sterility.

2

Brainstorm

PADMA PRIYA T R

Capture the incoming video	Use Deep learning algorithm	Convolutional Neural Network
UI design	Segmentation of images	IBM Watson Studio
Train the dataset	Image Processing	Recognition of Gestures

THANYA S S

Consider bad lighting	Jupyter Notebook	Video frames
Collect dataset	Deep Learning Algorithms	Flask Framework
Consider GPUs	Bug free	Pick appropriate gestures

NARESH K

Noise removal	Check lightings	Split training and testing images
Use CNN	IBM AI Studio	YOLO
ReLU Layer	Quality of input images	Computer Vision

PRANAV R

Use of DL Algorithms	OpenCV	Activation Functions
Feature Scaling	Tuning the model	Neural Networks
Python	Processing of images	Object detection

3

Group ideas

Image Processing

Capture the incoming video	Video frames	Object detection	Computer Vision	OpenCV

Deep learning model

Neural Networks	Convolution Neural Network	ReLU Layer	YOLO	Recognition of Gestures

User Interface

UI design	Flask Framework	Python

Platforms

Jupyter Notebook	IBM AI Studio	IBM Watson Studio

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

3.3 Proposed Solution
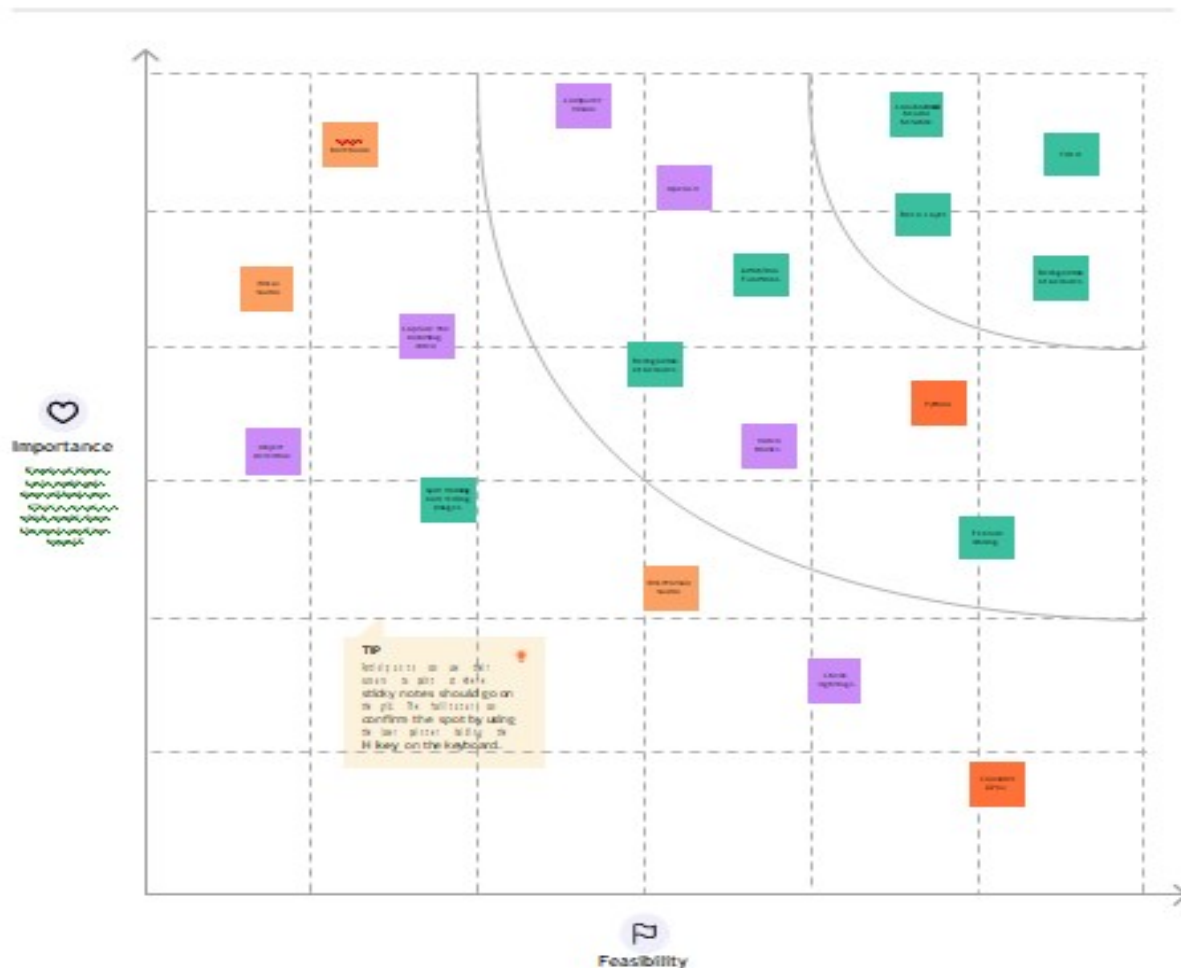
In our solution, Convolution Neural Network is used. First the model is trained pre trained on the images of different hand gestures, such as a showing numbers with fingers as 1,2,3,4. This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the gesture predicts is 1 then images is blurred; 2, image is resized; 3, image is rotated; 4, rectangle.

SI. No	Parameter	Description
1	Problem Statement (Problem to be solved)	Computer information technology is increasingly penetrating into the hospital domain. 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. Even though voice control also provides sterility, the noise level in the operating room (OR) deems it problematic. Our solution for this problem is the use of hand gestures as an alternative to existing interface techniques, offering the major advantage of sterility. In our work we refer to gestures as a basic form of non-verbal communication made with the hands.
2	Idea / Solution description	In this project Convolution Neural Network is used. 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 1 then images is blurred;2, image is resized;3,image is rotated; 4, rectangle.
3	Novelty / Uniqueness	The proposed system prevents surgeon's focus shift and change of location while achieving, rapid intuitive interaction with

		image databases. The system allows the surgeon to use his/her hands, their natural work tool. Non-verbal instructions by hand gesture commands used in this project are intuitive and fast.
4	Social Impact / Customer Satisfaction	This system assists surgeons while performing operations at a fast rate without any physical contact. Customers are highly benefited as the surgeries can be performed without touching any pointing devices. It also saves time. It can also be placed in other industries like banking. It can also help blind people.
5	Business Model (Revenue Model)	This system can be used in hospitals and diagnosis centers. It can also be placed in private and government medical camps.
6	Scalability of the Solution	More number of gestures can be added so that they can be improved. In addition to this, more number of images can be added so that the system makes correct prediction.

3.4   Problem Solution fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Our customers are medical professionals like Doctors, Nurses, Lab Technicians who wants optimized manipulation of the radiology images of the patients While performing surgeries.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

The tool may perform badly in poor lightings. The system is costly to setup as it requires expensive cameras and other supporting devices.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

"Gestix," a vision-based hand gesture capture and system that interprets in real-time the user's gestures for navigation and manipulation of images in an EMR Database is available in the market. The system is fast in making accurate predictions and it is easy to use. The drawback is that it can predict gestures only at a distance up to 5 meters from the camera.

Explore AS, differentiate

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Launching the model	Launch from the cloud where it is deployed Launch offline by installation
FR-2	Image Capturing	Capture Image via Camera Upload images from local system
FR-3	Perform gestures	Gestures should be captured correctly in the camera and proper gestures should be

FR No.		Description
		performed
FR-4	Model rendering	When the user captures/uploads the gestures followed by a click on the submit button, the deep learning algorithm should starts its processing task.
FR-5	Sterile browsing	After recognizing the gestures, sterile browsing of images should be done properly
FR-6	Displaying the images	The browsed images should be visible to the user.

4.2 Non- Functional requirement

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system designed is user-friendly
NFR-2	**Security**	The system is enforced with security mechanisms to avoid data theft.
NFR-3	**Reliability**	The system predicts the gestures and perform sterile browsing accurately and it produces improper results
NFR-4	**Performance**	It handles the visitors with a handsome response time, it is scalable and the underlying hardware and software is perfect.
NFR-5	**Availability**	The system is available or accessible by an authorized user whenever it is needed. It is not influenced by Denial of Service and Loss of Data Processing Capabilities
NFR-6	**Scalability**	The developed software model never downs the website due to an increase in website visitors.

5. PROJECT DESIGN
5.1 Data Flow Diagram

```
        ┌──────────────────────┐
        │   Open the software  │
        └──────────────────────┘
                   │
                   ▼
        ┌──────────────────────────┐
        │ Open Camera/ Upload Image │
        └──────────────────────────┘
                   │
                   ▼
        ┌──────────────────────┐◄──────────────┐
        │   Perform gestures   │                │
        └──────────────────────┘                │ No
                   │                            │
                   ▼                            │
            ╱──────────────╲                    │
           ╱    Gesture     ╲───────────────────┘
           ╲   Predicted?   ╱
            ╲──────────────╱
                   │  Yes
                   ▼
        ┌──────────────────────────┐
        │ Performs sterile browsing │
        └──────────────────────────┘
                   │
                   ▼
        ┌──────────────────────┐
        │    Display output    │
        └──────────────────────┘
                   │
                   ▼
            ┌──────────┐
            │   Exit   │
            └──────────┘
```
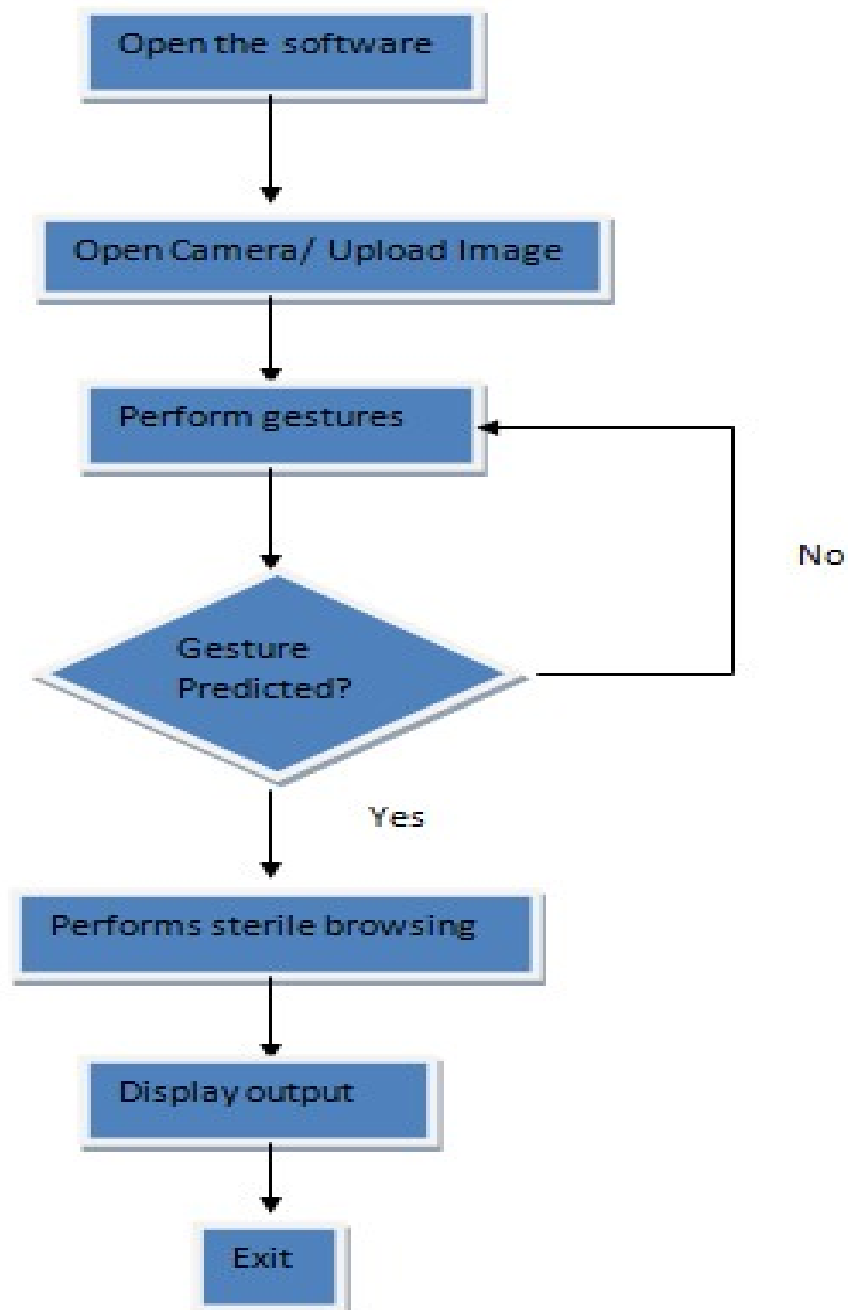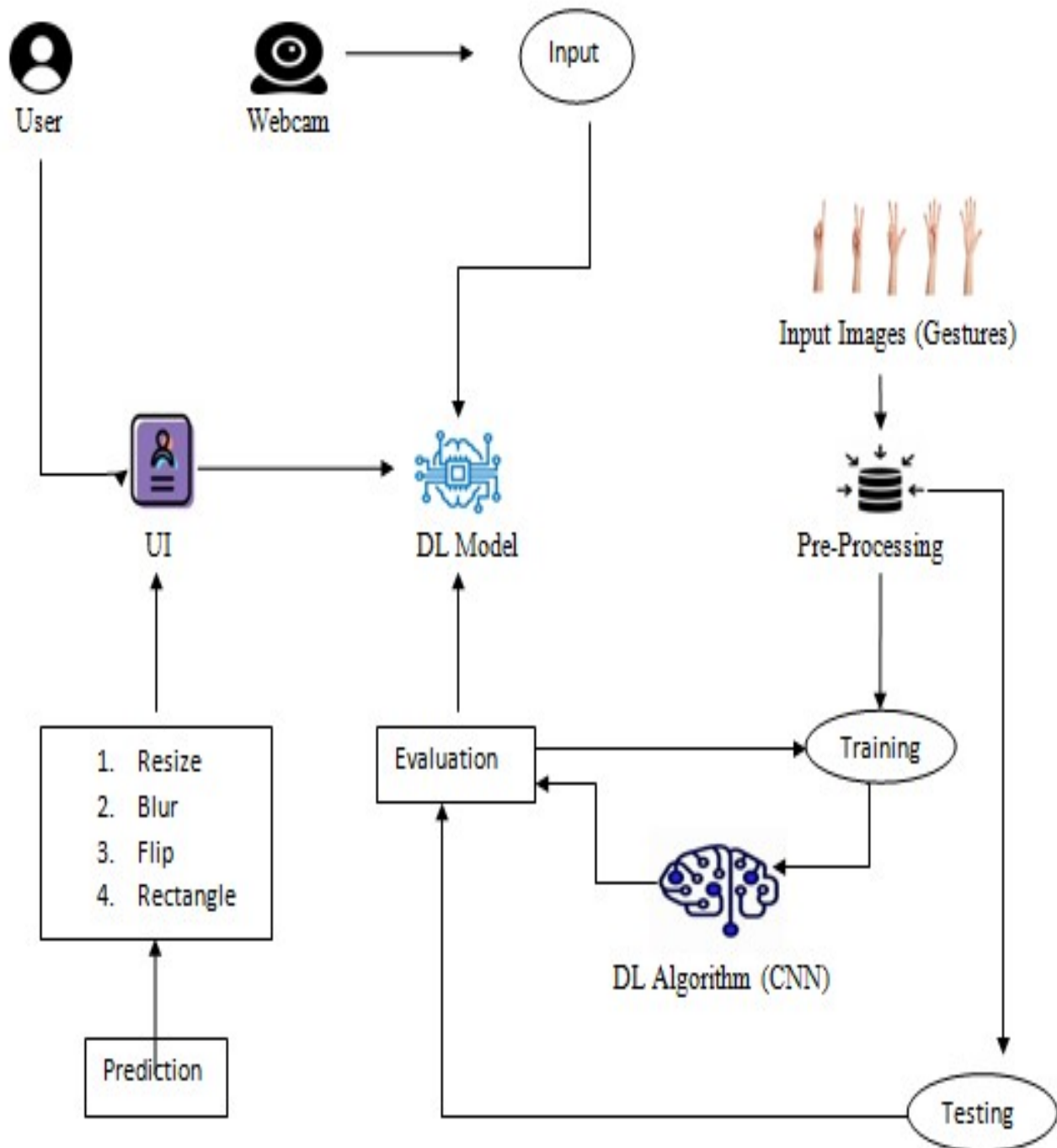
5.2 Solution & Technical Architecture
5.2.1 Solution Architecture
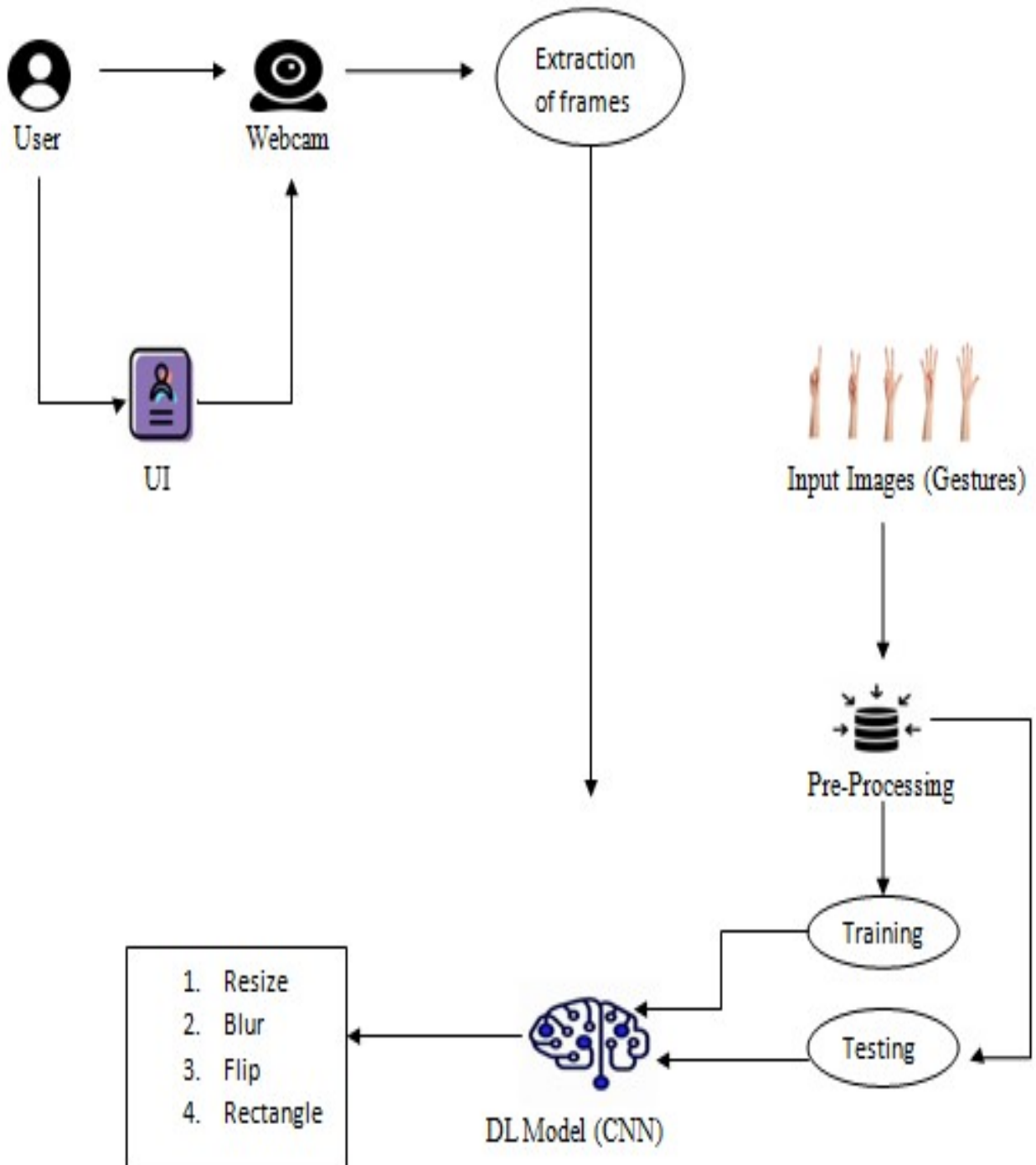
5.2.2 Technical Architecture

5.3 User Stories

The following depicts the user stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Surgeons	Launching Software	USN-1	As a user, I can launch the developed software.	I can launch the software	Low	Sprint-1
	Access UI	USN-2	As a user, I can use the software and operate on the UI	I can access the UI	Medium	Sprint-1
Surgeons (Performs live browsing using camera)	Launching camera	USN-3	As a user, I can open the camera from the software to perform gestures	I can open the camera to capture the images	Low	Sprint-2
Surgeons (Performs browsing by uploading existing images)	Upload images from local system	USN-4	As a user, I can upload images to the software from the local system	I can upload images to the software	Low	Sprint-2
Surgeons (Performs live browsing using camera)	Perform Gestures	USN-5	As a user, I can perform various gestures with respect to system specification for processing.	I can perform various gestures	Medium	Sprint-3
	Display	USN-6	As a user, I	I can see	High	Sprint-

	output		can see the sterile browsed image with respect to the gestures performed, displayed on the screen	the sterile browsed image on the screen		4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Launching Software	USN-1	As a user, I can launch the developed software.	1	Low	Padma Priya T R Thanya S S Naresh K Pranav R
Sprint-1	Access UI	USN-2	As a user, I can use the software and operate on the UI	1	Medium	Padma Priya T R Thanya S S Naresh K Pranav R
Sprint-2	Launching camera	USN-3	As a user, I can open the camera from the software to perform gestures	1	Low	Padma Priya T R Thanya S S Naresh K Pranav R
Sprint-2	Upload images from local system	USN-4	As a user, I can upload images to the	2	Low	Padma Priya T R Thanya S

			software from the local system			S Naresh K Pranav R
Sprint-3	Perform Gestures	USN-5	As a user, I can perform various gestures with respect to system specification for processing.	2	Medium	Padma Priya T R Thanya S S Naresh K Pranav R
Sprint-4	Display output	USN-6	As a user, I can see the sterile browsed image with respect to the gestures performed, displayed on the screen	2	High	Padma Priya T R Thanya S S Naresh K Pranav R

6.2 Sprint Delivery Schedule

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

Velocity:

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

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

6.3   Reports from JIRA
6.3.1 Velocity Report

Velocity report   › How to read this report

Commitment   Completed
The amount of work in the sprint when it began.   The amount of work done during the sprint.
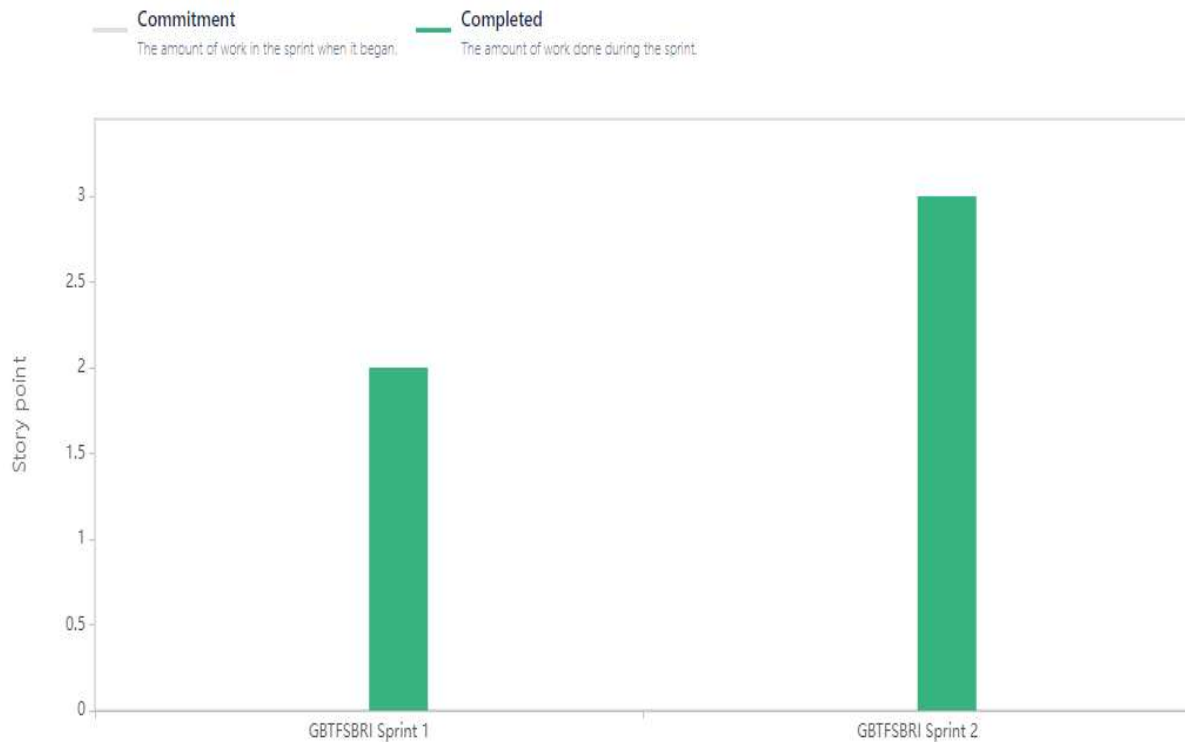
6.3.2 Sprint Burndown Chart

Date - October 24th, 2022 - October 29th, 2022
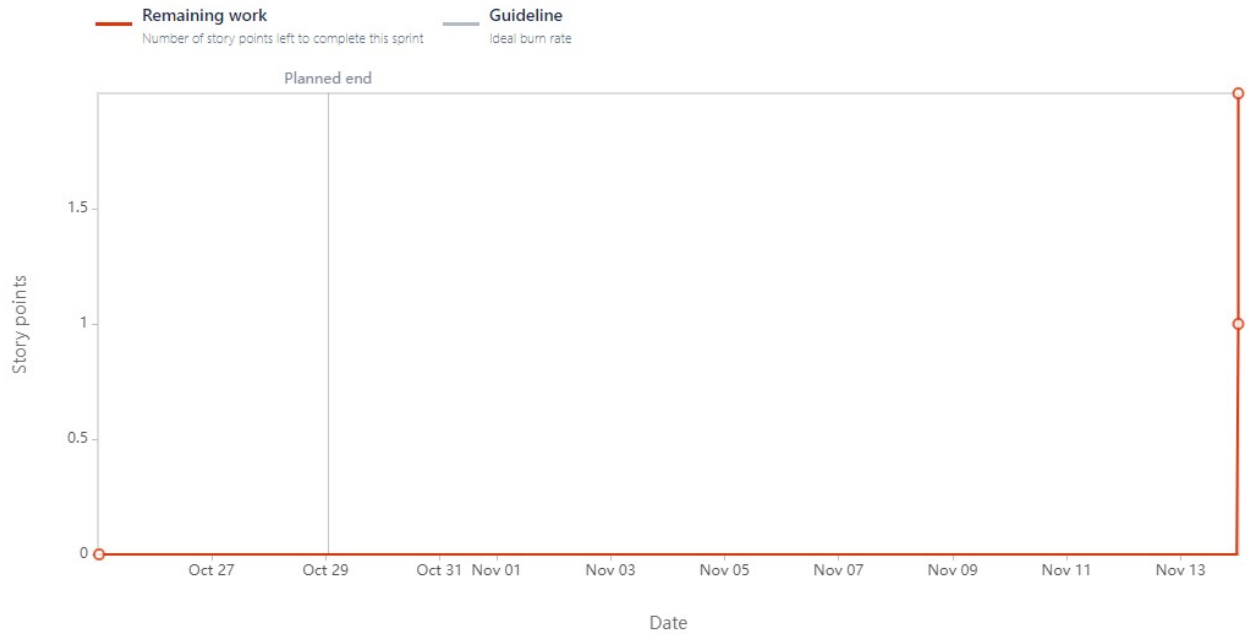
Date - October 31st, 2022 - November 5th, 2022
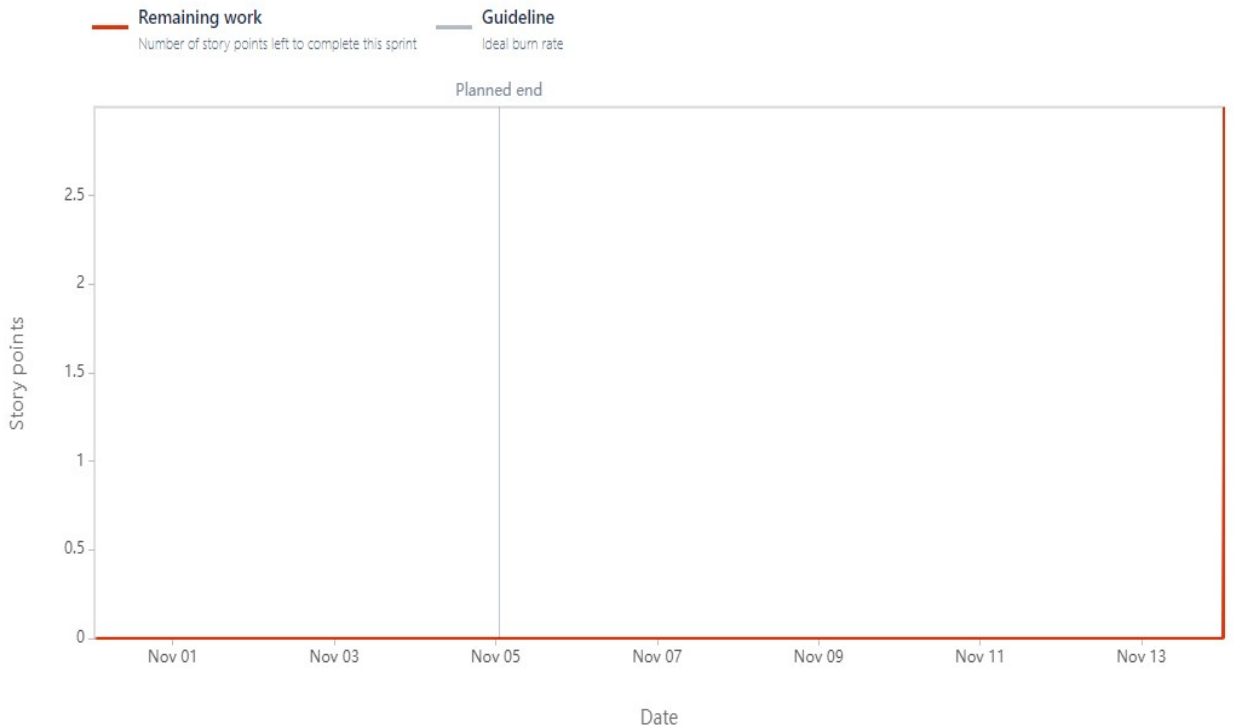
Date - November 7th, 2022 - November 12th, 2022

Date - November 14th, 2022 - November 19th, 2022

7. CODING & SOLUTIONING

7.1 Feature 1

The proposed solution is based on Convolutional neural network. A Convolutional neural network, or CNN, is a deep learning neural network designed for processing structured arrays of data such as images. Convolutional neural networks are widely used in computer vision and have become the state of the art for many visual applications such as image classification, and have also found success in natural language processing for text classification.

7.1.1 Import The ImageDataGenerator Library

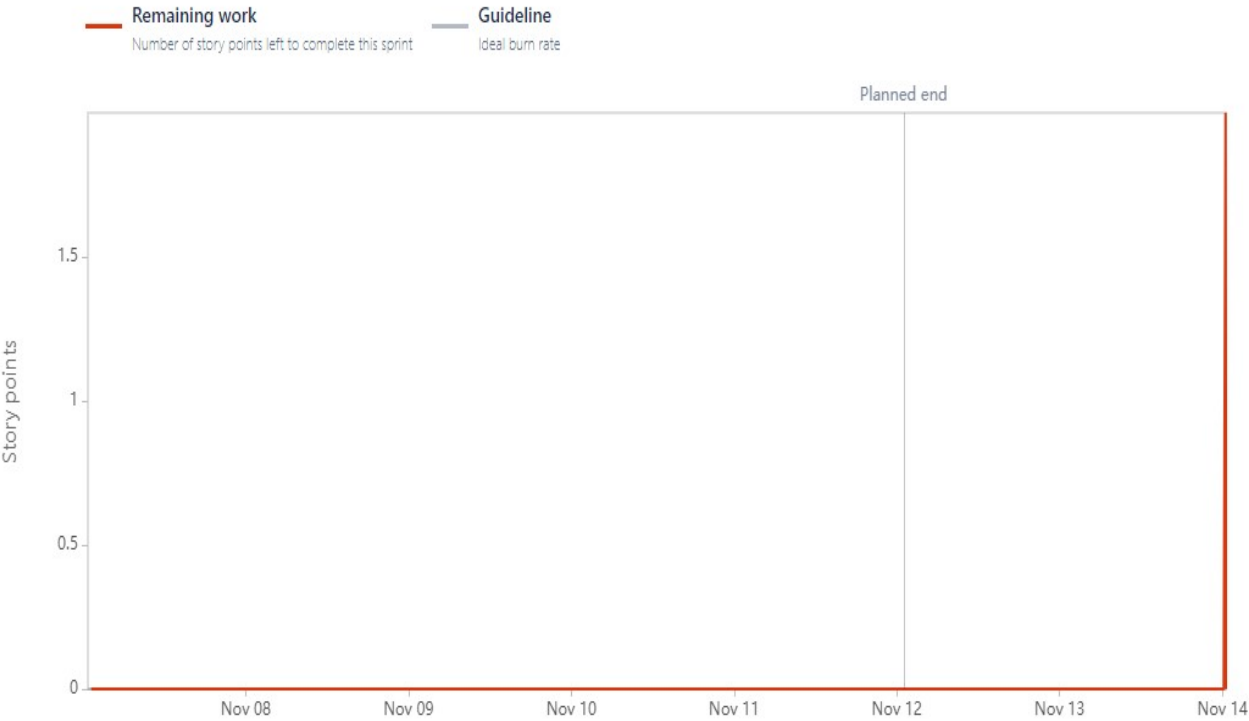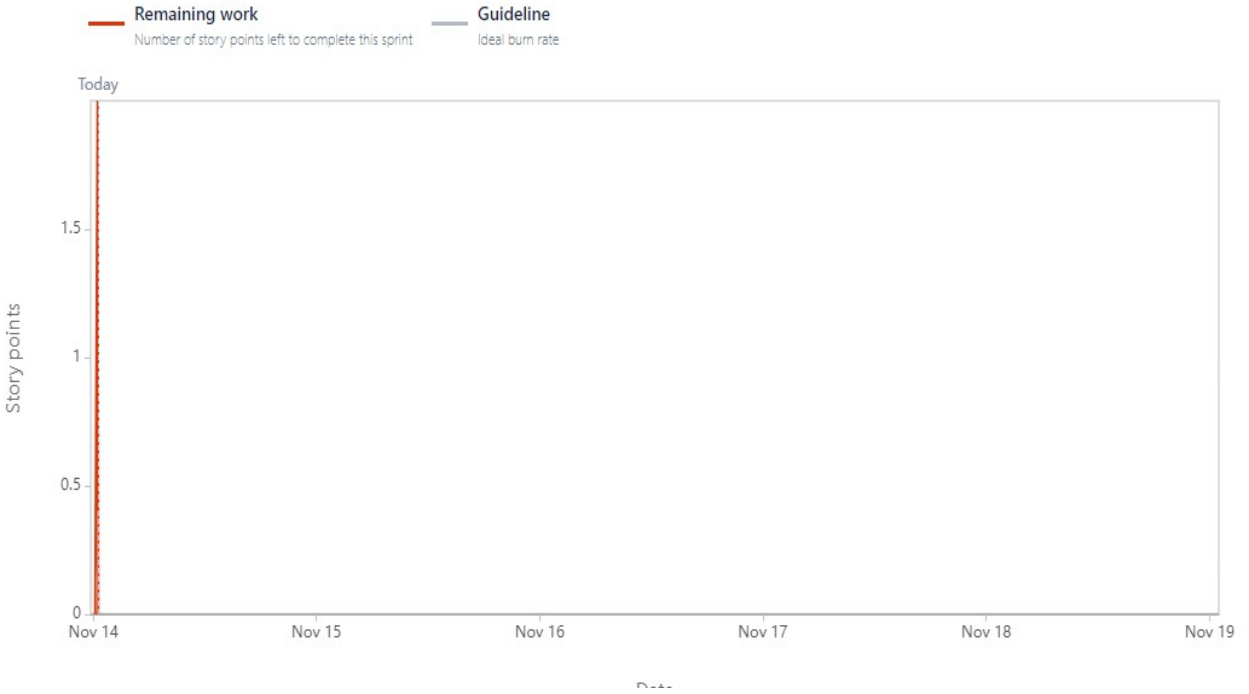Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class. The ImageDataGenerator class is imported from keras.

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

7.1.2 Configure ImageDataGenerator Class

ImageDataGenerator class is instantiated and the configuration for the types of data augmentation. There are five main types of data augmentation techniques for image data; specifically:

- Image shifts via the width_shift_range and height_shift_range arguments.
- Image flips via the horizontal_flip and vertical_flip arguments.
- Image rotations via the rotation_range argument.
- Image brightness via the brightness_range argument.
- Image zoom via the zoom_range argument.

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

```
test_datagen = ImageDataGenerator(rescale=1./255)
```

7.1.3 Apply ImageDataGenerator Functionality To Train set And Test set

To apply ImageDataGenerator functionality for Training set use flow_from_directory function. This function will return batches of images from the subdirectories 0,1,2,3,4,5 together with labels 0 to 5{'0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5}

Arguments:

directory: Directory where the data is located. If labels is "inferred", it should contain subdirectories, each containing images for a class. Otherwise, the directory structure is ignored.

batch_size: Size of the batches of data. Default: 32.

target_size: Size to resize images to after they are read from disk.

class_mode:

- 'int': means that the labels are encoded as integers (e.g. for sparse_categorical_crossentropy loss).

- 'categorical' means that the labels are encoded as a categorical vector (e.g. for categorical_crossentropy loss).

- 'binary' means that the labels (there can be only 2) are encoded as float32 scalars with values 0 or 1 (e.g. for binary_crossentropy).

 - None (no labels).

ImageDataGenerator functionality to Trainset and Testset by using the following code

```
train_data= train_datagen.flow_from_directory (r 'D:\Dataset\train', target_size=(128,128),
batch_size=8, class_mode='categorical', subset='training', color_mode= 'grayscale')
```

```
test_data = test_datagen.flow_from_directory(r'D:\Dataset\test',
target_size=(128,128),batch_size=8,class_mode='categorical',color_mode='grayscale')
```

7.1.4 Importing The Model Building Libraries

Import the necessary libraries required for model building

```python
import tensorflow as tf
import keras
import numpy as np
from tensorflow.keras import layers,losses
from tensorflow.keras.models import Sequential, Model, load_model
from tensorflow.keras.models import Sequential, Model, load_model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras import regularizers
from keras.layers import BatchNormalization
```

7.1.5 Initializing The Model

Sequential model is a linear stack of layers. We can create a Sequential model by passing a list of layer instances to the constructor: from keras.models import Sequential from keras as follows.

```python
model = Sequential()
```

7.1.6 Adding CNN Layers

We are adding a convolution layer with activation function as "relu" and with a small filter size (3,3) and number of filters (32) followed by a max pooling layer. Max pool layer is used to down sample the input. Flatten layer flattens the input. It does not affect the batch size.

```python
model.add(BatchNormalization(input_shape = (128,128,1)))
model.add(Convolution2D(32, (3,3), activation ='relu', input_shape = (128, 128, 1)))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=6,kernel_size=4,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=3,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=2,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(MaxPooling2D(pool_size=2))
```

7.1.7 Adding Dense Layers

Dense layer is deeply connected neural network layer. It is most common and frequently used layer.

```
model.add(Dense(units=128,activation = 'relu'))
model.add(Dense(units=64,activation = 'relu'))
model.add(Dense(units=32,activation = 'relu'))
model.add(Dense(units = 6, activation = 'softmax'))
```

Understanding the model is very important phase to properly use it for training and prediction purposes. Keras provides a simple method, summary to get the full information about the model and its layers. The following code represents that.

model.summary()

7.1.8 Compiling the model

The compilation is the final step in creating a model. Once the compilation is done, we can move on to training phase. Loss function is used to find error or deviation in the learning process. Keras requires loss function during model compilation process. Optimization is an important process which optimizes the input weights by comparing the prediction and the loss function. Here we are using adam optimizer. Metrics is used to evaluate the performance of your model. It is similar to loss function, but not used in training process.

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

7.1.9 Train the model on IBM

Next, the process of training the model with image dataset is done. fit_generator functions used to train a deep learning neural network

Arguments:

steps_per_epoch: it specifies the total number of steps taken from the generator as soon as one epoch is finished and next epoch has started. We can calculate the value of steps_per_epoch as the total number of samples in your dataset divided by the batch size.

Epochs: an integer and number of epochs we want to train our model for.

validation_data can be either:

- an inputs and targets list

- a generator
- an inputs, targets, and sample_weights list which can be used to evaluate
- the loss and metrics for any model after any epoch has ended.

validation_steps: only if the validation_data is a generator then only this argument can be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.

The model is trained on IBM Cloud platform

```
model.fit_generator(train_gen,epochs=25,
steps_per_epoch=18000//32,steps_per_epoch=18000//32, validation_steps=3600//32)
```

7.1.10 Save the model

The model is saved with .h5 extension as follows. An H5 file is a data file saved in the Hierarchical Data Format (HDF). It contains multidimensional arrays of scientific data.

```
model.save(r'D:\IBM Project\gesture.h5')
```

7.2 Feature 2

7.2.1 Create HTML Pages

We use HTML to create the front end part of the web page. Here, we created 3 html pages- Home.html, Intro.html and Index.html. Home.html displays home page. Intro.html displays introduction about the hand gesture recognition. Index.html accepts input from the user (live video) and predicts the values. We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML pages.

7.2.2 Build Python Code

Flask is a web framework written in python for server-side scripting. The step by step procedure for building the backend application is given below.

MediaPipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform Framework works in Desktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano.

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.

MediaPipe is used detect the hand using the co-ordinates from the live video and passes this as input to the deep learning model.

The deep learning model after getting the input from camera processes it and predicts the output. With respect to the predicted gesture, Resize, Blur, Rotation and Rectangle operations are done.

The following code illustrates the overall process

```python
import mediapipe as mp
import numpy as np
from flask import Flask, render_template, request
import cv2
import os
from keras.models import load_model
app = Flask(__name__)
@app.route("/")
def home():
    return render_template("home.html")

@app.route("/process", methods=['GET', 'POST'])
def process():
    if request.method == 'POST':
        upload_image = request.files['upload_image']
        img=upload_image.read()
        npimg = np.fromstring(img, np.uint8)
```

```python
    model1 = load_model('gesture.h5')
    mpHands = mp.solutions.hands
    hands = mpHands.Hands(max_num_hands=1, min_detection_confidence=0.5,
min_tracking_confidence=0.5)
    mpDraw = mp.solutions.drawing_utils
    cap = cv2.VideoCapture(0)
    while True:
        _, frame = cap.read()

        h, w, c = frame.shape

        frame = cv2.flip(frame, 1)
        framergb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        result = hands.process(framergb)
        res = ''

        if result.multi_hand_landmarks:
            landmarks = []
            for handslms in result.multi_hand_landmarks:
                x_max = 0
                y_max = 0
                x_min = w
                y_min = h
                for lm in handslms.landmark:
                    x = int(lm.x * w)
                    y = int(lm.y * h)

                    landmarks.append([x, y])
                    if x > x_max:
                        x_max = x
                    if x < x_min:
                        x_min = x
                    if y > y_max:
                        y_max = y
                    if y < y_min:
                        y_min = y
```

```python
            cv2.rectangle(frame, (x_min - 5, y_min - 5), (x_max + 5, y_max + 5), (0,
255, 0), 2)
            framegray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            hand = framegray[y_min - 5:y_max + 5, x_min - 5:x_max + 5]
            hand = cv2.resize(hand, (128, 128))
            hand = hand / 255
            hand = hand.reshape(128, 128, 1)
            hand = np.expand_dims(hand, axis=0)
            mpDraw.draw_landmarks(frame, handslms,
mpHands.HAND_CONNECTIONS)
            prediction = model1.predict(hand)
            res = np.argmax(prediction)
            image1 = cv2.imdecode(npimg, cv2.IMREAD_COLOR)
            image1= cv2.resize(image1, (400, 400))
            if res==1:
                resized = cv2.resize(image1, (200, 200))
                cv2.imshow("Resizing", resized)
                key=cv2.waitKey(3000)

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

            elif res==2:
                blurred = cv2.GaussianBlur(image1, (21, 21), 0)
                cv2.imshow("Blurred", blurred)
                key=cv2.waitKey(3000)
                if (key & 0xFF) == ord("3"):
                    cv2.destroyWindow("Blurred")

            elif res==3:
                (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"):
```

```python
                cv2.destroyWindow("OpenCV Rotation")


            elif res==4 :
                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")


            else:
                continue

        #cv2.putText(frame, str(res), (10, 50), cv2.FONT_HERSHEY_SIMPLEX,
            #1, (0, 0, 255), 2, cv2.LINE_AA)
        cv2.imshow("Output", frame)

        if cv2.waitKey(1) == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

  return render_template("index.html")


@app.route("/intro")
def intro_page():
  return render_template("intro.html")


@app.route("/index")
def index_page():
  return render_template("index.html")


@app.route("/back")
```

```
def back():
    return render_template("home.html")
```

7.2.3 Run the application

At last, the flask application is run using the following command.

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

8. TESTING

8.1 Test Cases

Test Case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps to execute	Test Data	Expected Result	Actual Result	Status
Home Page _TC _OO 1	Functional	Home Page	Verify user is able to see the home page when user clicked on the application link	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2.Verify whether the home page of the application displayed or not	Application Url	Home Page should be displayed	Working as expected	Pass
Home Page _TC _OO 2	UI	Home Page	Verify the UI elements in Home Page	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Verify the UI Component	Application Url	Application should show the 'Get Started'	Button is not visible	Fail

					s of the home page by checking the 'Get Started' button.		button		
Hom Page _TC _OO 3	UI	Ho me Pag e	Verify the UI elements in Home Page	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Verify the UI Component s of the home page by checking the'Get Started' button.	App licat ion Url	Applic ation should show the 'Get Starte d' button	Worki ng as expect ed	Pass
Hom Page _TC _OO 4	Funct ional	Ho me Pag e	Verify whether the user is able to move to intro page or not?	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button 3. Verify whether the user is able to move to intro page or not?	App licat ion Url	User should naviga te to intro page	Not naviga ted to intro page	Fail
Hom Page _TC	Funct ional	Ho me Pag	Verify whether the user is	1. Laptop 2.Internet Connection	1. Enter URL in the browser	App licat ion	User should naviga	Worki ng as expect	Pass

_OO 5		e	able to move to intro page or not?	3. Web cam	and click enter. 2.Click on the 'Get Started' button 3. Verify whether the user is able to move to intro page or not?	Url	te to intro page	ed	
Intro Page _TC _OO 1	Funct ional	Intr o Pag e	Verify user is able to open the intro page	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2.Click on the 'Get Started' button 3. Verify whether the user is able to open the intro page or not?	App licat ion Url	Intro page should be opene d	Worki ng as expect ed	Pass
Intro Page _TC _OO 2	UI	Intr o Pag e	Verify the UI elements in Intro Page	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3.Verify the following UI	App licat ion Url	UI compo nents should be displa yed on the screen .	UI compo nents not displa yed	Fail

					Components of the intro page a. Intro Text b. Proceed button c. Back button				
Intro Page _TC _OO 3	UI	Intr o Pag e	Verify the UI elements in Intro Page	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3.Verify the following UI Components of the intro page a. Intro Text b. Proceed button c. Back button	App licat ion Url	UI compo nents should be displa yed on the screen .	Worki ng as expect ed	Pass
Intro Page _TC _OO 4	Funct ional	Intr o Pag e	Verify whether the user is able to navigate to index page	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button.	App licat ion Url	Index page should be opene d	Worki ng as expect ed	Pass

					3. Click on 'Proceed' button. 4.Verify whether the index page is opening or not				
Intro Page_TC_OO5	Functional	Intro Page	Verify whether the user is able to navigate back to the home page	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on 'Back' button. 4.Verify whether the home page is opening or not	Application Url	Home Page should be displayed	Home page not opened	Fail
Intro Page_TC_OO6	Functional	Intro Page	Verify whether the user is able to navigate back to the home page	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on 'Back' button. 4.Verify whether the home page	Application Url	Home Page should be displayed	Working as expected	Pass

					is opening or not				
Inde xPag e_T C_O O1	Funct ional	Inde x Pag e	Verify whether the user is able to open the index page or not	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on 'Proceed' button. 4.Verify whether the index page is opening or not	App licat ion Url	Index page should be opene d	Worki ng as expect ed	Pass
Inde xPag e_T C_O O2	UI	Inde x Pag e	Verify the UI elements in Intro Page	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on the 'Proceed' button. 4. Verify the following UI Component s of the index page a. File	App licat ion Url	UI compo nents should be displa yed on the screen .	Worki ng as expect ed	Pass

					upload button b. Video Capture button c. Image display field d. Back button				
IndexPage_TC_OO3	Functional	Index Page	Verify whether the user is able to upload the image or not.	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on the 'Proceed' button. 4. Click on the upload file button. 5. Choose an image from gallery. 6. Upload the image. 7. Verify whether the image is uploaded or not.	Image from local system	Image should be uploaded	Image not uploaded	Fail
IndexPage_T e_T	Functional	Index Pag	Verify whether the user is	1. Laptop 2.Internet Connection	1. Enter URL in the browser	Image fro	Image should be	Working as expect	Fail

Test Case ID	Type	Page	Description	Prerequisites	Steps	Test Data	Expected Result	Actual Result	Status
C_O O4		e	able to upload the image or not.	3. Web cam	and click enter. 2. Click on the 'Get Started' button. 3. Click on the 'Proceed' button. 4. Click on the upload file button. 5. Choose an image from gallery. 6. Upload the image. 7. Verify whether the image is uploaded or not.	m loca l syst em	upload ed	ed	
Inde xPag e_T C_O O5	Funct ional	Inde x Pag e	Verify whether the user uploaded image is displayed on the screen or not.	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on the 'Proceed' button. 4. Click on the upload file button.	Ima ge fro m loca l syst em	Image should be displa yed on the screen .	Worki ng as expect ed	Fail

					5. Choose an image from gallery. 6. Upload the image. 7. Verify whether the image is displayed on the screen or not.				
Inde xPag e_T C_O O6	Funct ional	Inde x Pag e	Verify whether the system is capturing live video	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on the 'Proceed' button. 4. Click on the capture video button. 5. Verify whether the live video is captured or not.	Liv e Vid eo	Live video should be captur ed	Worki ng as expect ed	Fail
Inde xPag e_T C_O O7	Funct ional	Inde x Pag e	Verify whether the model predicts the	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter.	Liv e Vid eo	Gestur e should be predic	Wron g predic tion	Fail

			correct gesture		2. Click on the 'Get Started' button. 3. Click on the 'Proceed' button. 4. Click on the capture video button. 5. Verify whether the system predicts the correct gesture or not.		ted correc tly.		
Inde xPag e_T C_O O8	Funct ional	Inde x Pag e	Verify whether the model predicts the correct gesture	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on the 'Proceed' button. 4. Click on the capture video button. 5. Verify whether the system predicts the	Liv e Vid eo	Gestur e should be predic ted correc tly.	Worki ng as expect ed	Pass

							correct gesture or not.		
Inde xPag e_T C_O O9	Funct ional	Inde x Pag e	Verify whether the system is performin g the correct task with respect to the gesture	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on the 'Proceed' button. 4. Click on the capture video button. 5.Verify whether the system is performing the correct task with respect to the gesture	Liv e Vid eo	Task should be perfor med with respec t to the gestur e.	Wron g gestur e	Fail
Inde xPag e_T C_O O10	Funct ional	Inde x Pag e	Verify whether the system is performin g the correct task with respect to the gesture	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on the 'Proceed'	Liv e Vid eo	Task should be perfor med with respec t to the gestur e.	Worki ng as expect ed	Pass

					button. 4. Click on the capture video button. 5.Verify whether the system is performing the correct task with respect to the gesture				
Inde xPag e_T C_O O11	Funct ional	Inde x Pag e	Verify whether the user is able to quit the screen after performin g tasks	1. Laptop 2.Internet Connection 3. Web cam	1. Enter URL in the browser and click enter. 2. Click on the 'Get Started' button. 3. Click on the 'Proceed' button. 4. Click on the capture video button. 5.Analyze the task performed 6.Press 'q' to quit	Pres sing 'q'	Camer a screen should be closed	Worki ng as expect ed	Pass

8.2 User Acceptance Testing

8.2.1 Purpose of Document

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

8.2.2 Defect Analysis

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

Resolution	Severity1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	1	2	0	3
Duplicate	0	0	0	0	0
External	2	1	1	0	4
Fixed	2	2	3	0	7
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	4	4	6	0	14

8.2.3 Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Home Page	5	0	2	3
Intro Page	6	0	2	4
Index Page	11	0	3	8

9. RESULTS

9.1 Performance Metrics

The performance of the solution is measured in terms of validation accuracy and the loss in every epoch. The model summary and the accuracy is given as follows.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	Model: "sequential" Layer (type) — Output Shape — Param # batch_normalization (BatchNormalization) — (None, 128, 128, 1) — 4 conv2d (Conv2D) — (None, 126, 126, 32) — 320 max_pooling2d (MaxPooling2D) — (None, 63, 63, 32) — 0 conv2d_1 (Conv2D) — (None, 63, 63, 6) — 3078 max_pooling2d_1 (MaxPooling2D) — (None, 31, 31, 6) — 0 conv2d_2 (Conv2D) — (None, 31, 31, 128) — 7040 max_pooling2d_2 (MaxPooling2D) — (None, 15, 15, 128) — 0 conv2d_3 (Conv2D) — (None, 15, 15, 128) — 65664 max_pooling2d_3 (MaxPooling2D) — (None, 7, 7, 128) — 0 flatten (Flatten) — (None, 6272) — 0 dense (Dense) — (None, 128) — 802944 dense_1 (Dense) — (None, 64) — 8256 dense_2 (Dense) — (None, 32) — 2080 dense_3 (Dense) — (None, 6) — 198 Total params: 889,584 Trainable params: 889,582 Non-trainable params: 2

| 2. | Accuracy | Training Accuracy – 99.92%

 Validation Accuracy – 99.86% | <pre>Epoch 1/20
562/562 [==============================] - 308s 546ms/step - loss: 0.2178 - accuracy: 0.9182 - val_loss: 0.0014 - val_accuracy: 1.0000
Epoch 2/20
562/562 [==============================] - 306s 545ms/step - loss: 0.0386 - accuracy: 0.9880 - val_loss: 0.0068 - val_accuracy: 0.9978
Epoch 3/20
562/562 [==============================] - 299s 532ms/step - loss: 0.0192 - accuracy: 0.9948 - val_loss: 7.7184e-04 - val_accuracy: 1.0000
Epoch 4/20
562/562 [==============================] - 306s 545ms/step - loss: 0.0137 - accuracy: 0.9960 - val_loss: 0.0032 - val_accuracy: 0.9983
Epoch 5/20
562/562 [==============================] - 310s 552ms/step - loss: 0.0184 - accuracy: 0.9940 - val_loss: 5.2546e-04 - val_accur
Epoch 6/20
562/562 [==============================] - 312s 556ms/step - loss: 0.0150 - accuracy: 0.9949 - val_loss: 1.4421e-05 - val_accuracy: 1.0000
Epoch 7/20
562/562 [==============================] - 304s 541ms/step - loss: 0.0073 - accuracy: 0.9978 - val_loss: 0.0120 - val_accuracy: 0.9967
Epoch 8/20
562/562 [==============================] - 309s 550ms/step - loss: 0.0111 - accuracy: 0.9966 - val_loss: 0.0011 - val_accuracy: 0.9994
Epoch 9/20
562/562 [==============================] - 313s 556ms/step - loss: 0.0067 - accuracy: 0.9982 - val_loss: 2.7600e-05 - val_accuracy: 1.0000
Epoch 10/20
562/562 [==============================] - 311s 553ms/step - loss: 0.0124 - accuracy: 0.9962 - val_loss: 4.3617e-04 - val_accuracy: 1.0000
Epoch 11/20
562/562 [==============================] - 311s 552ms/step - loss: 0.0071 - accuracy: 0.9978 - val_loss: 2.4401e-04 - val_accuracy: 1.0000
Epoch 12/20
562/562 [==============================] - 310s 552ms/step - loss: 0.0035 - accuracy: 0.9992 - val_loss: 0.0021 - val_accuracy: 0.9989
Epoch 13/20
562/562 [==============================] - 303s 539ms/step - loss: 0.0090 - accuracy: 0.9975 - val_loss: 0.0017 - val_accuracy: 1.0000
Epoch 14/20
562/562 [==============================] - 299s 533ms/step - loss: 0.0082 - accuracy: 0.9976 - val_loss: 1.9254e-05 - val_accuracy: 1.0000
Epoch 15/20
562/562 [==============================] - 308s 549ms/step - loss: 0.0072 - accuracy: 0.9982 - val_loss: 1.5410e-04 - val_accuracy: 1.0000
Epoch 16/20
562/562 [==============================] - 307s 545ms/step - loss: 0.0078 - accuracy: 0.9974 - val_loss: 1.8354e-04 - val_accuracy: 1.0000
Epoch 17/20
562/562 [==============================] - 306s 545ms/step - loss: 0.0020 - accuracy: 0.9996 - val_loss: 1.1681e-05 - val_accuracy: 1.0000
Epoch 18/20
562/562 [==============================] - 307s 546ms/step - loss: 0.0074 - accuracy: 0.9981 - val_loss: 1.8610e-04 - val_accuracy: 1.0000
Epoch 19/20
562/562 [==============================] - 300s 534ms/step - loss: 0.0051 - accuracy: 0.9986 - val_loss: 1.1064e-04 - val_accuracy: 1.0000
Epoch 20/20
562/562 [==============================] - 304s 540ms/step - loss: 0.0026 - accuracy: 0.9992 - val_loss: 0.0037 - val_accuracy: 0.9986</pre> |

10.   ADVANTAGES & DISADVANTAGES
10.1 Advantages

The main advantage of the system is that it prevents surgeon's focus shift and change of location while achieving, rapid intuitive interaction with image databases. The system allows the surgeon to use his/her hands, their natural work tool. Non-verbal instructions by hand gesture commands used in this project are

intuitive and fast. This system assists surgeons while performing operations at a fast rate without any physical contact. Customers are highly benefited as the surgeries can be performed without touching any pointing devices. It also saves time. It can also be placed in other industries like banking. It can also help blind people.

10.2 Disadvantages

The disadvantage of this system is that the system is quite expensive as it requires expensive cameras and other devices to capture images and process it

11. CONCLUSION

In this project we developed a tool which recognizes 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 user's 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 usabilityfor 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

Source Code: [GitHub - IBM-EPBL/IBM-Project-39149-1660398111: A Gesture-based Tool for Sterile Browsing of Radiology Images](#)

Project Demo Link: https://youtu.be/i0_kQM2omy4