Gesture Based Tool for Sterile Browsing of Radiology Images

TEAM ID:
PNT2022TMID28409

TEAM LEAD:
Mereshiya J M – 312419205057

MEMBERS:
Kavya P - 312419205044
Gayathri R - 312419205026
Jenitta Raja Kumari P - 312419205037

1.INTRODUCTION

Recent developments in computer software and related hardware technology have provided a value added service to the users. In everyday life, physical gestures are a powerful means of communication. They can economically convey a rich set of facts and feelings. For example, waving one's hand from side to side can mean anything from a "happy goodbye" to "caution". Use of the full potential of physical gesture is also something that most human computer dialogues lack .The task of hand gesture recognition is one the important and elemental problem in computer vision. With recent advances in information technology and media, automated human interactions systems are build which involve hand processing task like hand detection, hand recognition and hand tracking.This prompted my interest so I planned to make a software system that could recognize human gestures through computer vision, which is a sub field of artificial intelligence. The purpose of my software through computer vision was to program a computer to "understand" a scene or features in an image.A first step in any hand processing system is to detect and localize hand in an image. The hand detection task was however challenging because of variability in the pose, orientation, location and scale. Also different lighting conditions add further variability.

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.

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. The image of the gesture captured in the video frame is compared with the Pre-trained model and thegesture is identified. If the gesture predicts is 0 - then images is converted into rectangle, 1 -image is Resized into (200,200), 2 - image is rotated by -45°, 3 - image is blurred, 4 – image. In two brain surgeries at the Neurosurgery, procedures were observed by the authors to gain insights about the use of current technologies and how they affect the quality of the surgeon‘s performance. We found that: (a) surgeons kept their focus of attention between the patient and the surgical point of interest on the touch-screen navigation system; (b) a short distance between the surgeon and the patient was maintained during most of the surgery; (c) the surgeon had to move close to the main control wallto discuss and browse through the patient‘s MRI images

2. LITERATURE SURVEY

2.1 Existing Problem

Hand gesture recognition system confronts many challenges as addressed in , these challenges are: Variation of illumination conditions; where any change in the lighting condition affects badly on the extracted hand skin region . Rotation problem: this problem arises when the hand region rotated in any direction in the scene . Background problem; refers to the complex background where there is other objects in the scene with the hand objects and these objects might contain skin like color which would produce misclassification problem. Scale problem; this problem appears when the hand poses havedifferent sizes in the gesture image. Finally, Translation problem; the variation of hand positions in different images also leads to erroneous representation of the features.

2.2 References

2.2.1 A Gesture-based Tool for Sterile Browsing of Radiology Images - research paperby national library of medicine

The hand gesture control system –*Gestix*‖ developed by the authors helped the doctor to remainin place during the entire operation, without any need to move to the main control wall since allthe 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 IIvideo-capturing device is used.

2.2.2 Hatice Gunes, Massimo Piccardi, Tony Ja, 2007, Research School of Information Sciences and Engineering Australian National University Face and Body Gesture Recognition for a Vision-Based Multimodal Analyzer

For the computer to interact intelligently with human users, computers should be able to recognize emotions, by analyzing the human's affective state, physiology and behavior. In this paper, we present a survey of research conducted on face and body gesture and recognition. In order to make human-computer interfaces truly natural, we need to develop technology that tracks human movement, body behavior and facial expression, and interprets these movements in an affective way. Accordingly in this paper, we present a framework for a vision-based multimodal analyzer that combines face and body gesture and further discuss relevant issues.

2.2.3 J.Jenkinwinston (96207106036), M.Maria Gnanam (96207106056), R.Ramasamy (96207106306), Anna University of Technology, Tirunelveli: Hand Gesture Recognitionsystem Using Haar Wavelet.

Nowadays, sign language is commonly used as a communication language for auditory handicapped people. In addition to voice and controller pads, hand gestures can also be an effective way of communication between humans and robots or even between auditory handicapped people and robots. To be an effective sign recognition system, it should be glove-free, fast, small database and accurate. In this project, we aim to propose a hand gesture recognition system which performs realtime recognition. We reduce the database size by standardizing the orientation of hands using the idea of principal axis. To facilitate the matching process, we propose a codeword scheme based on the features of hand gestures.

2.3 PROBLEM STATEMENT

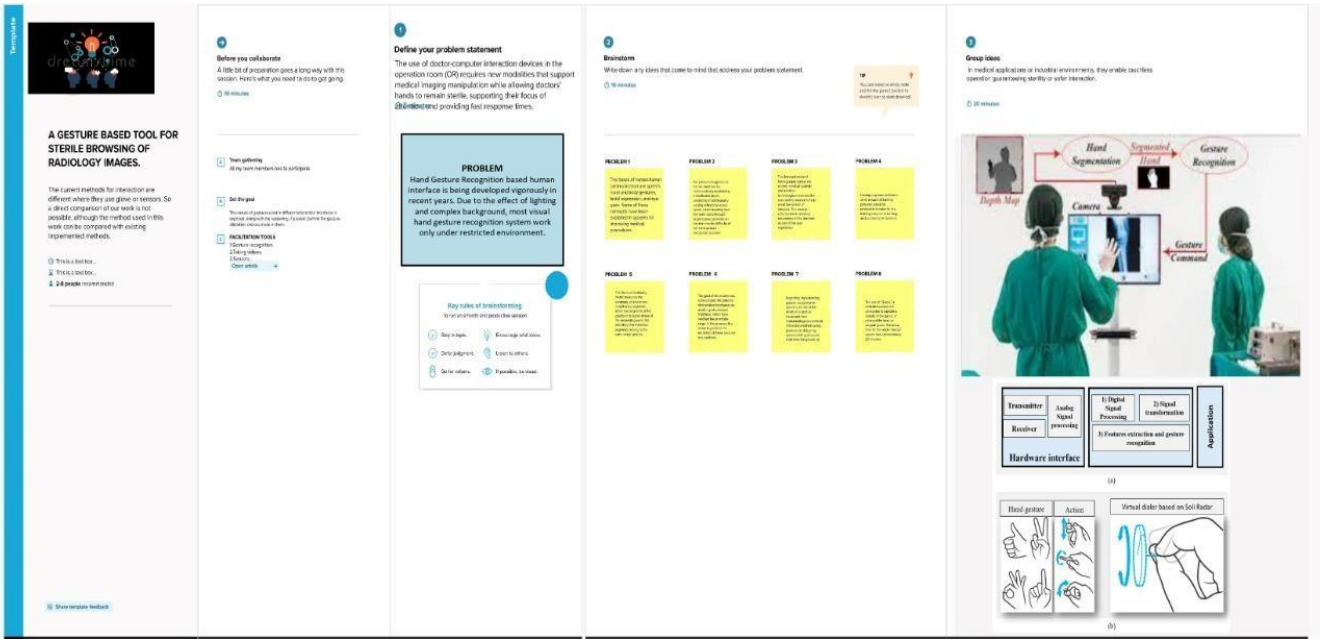
In this project we have used Convolutional Neural Network to frst train the model on the images of different hand gestures, like showing numbers with fngers as 0,1,2,3,4,5. Then we made a web portal using Flask where user can input any image on which he wants to perform the operations. After uploading the image, our portal uses the integrated webcam to capture the video frame using OpenCV. The gesture captured in the video frame is compared with the Pre-trained model and the gesture is identifed. If the prediction is 0 - then images is converted into rectangle, 1 - image is blurred , 2 - image is rotated by -45°, 3 - image is resized in (400,400) , 4 - image is Resized in (200,200) , 5 - image is converted into grayscale, but in real time we use of doctor-computer interaction devices in the operation room (OR) requires new modalities that support medical imaging manipulation while allowing doctors' hands to remain sterile, supporting their focus of attention, and providing fast response time.

3. IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHYMAP CANVAS:

SAYS This allows uses hand to remain sterile This supports the focus of attention which yields faster response at times	**DOES** By using convolution Neural Network train the model of the image by using different hand gestures Then the gesture is trained and compared with pre-defined model
THINKS Why to have a control over infotainment system without touching any buttons on the screen? Can computes be able to understand human body language?	**FEELS** Moment of capturing hand gestures and works with recognition system Interprets in real time for navigation and manipulation of images in a database.

3.2 IDEATION AND BRAINSTORMING

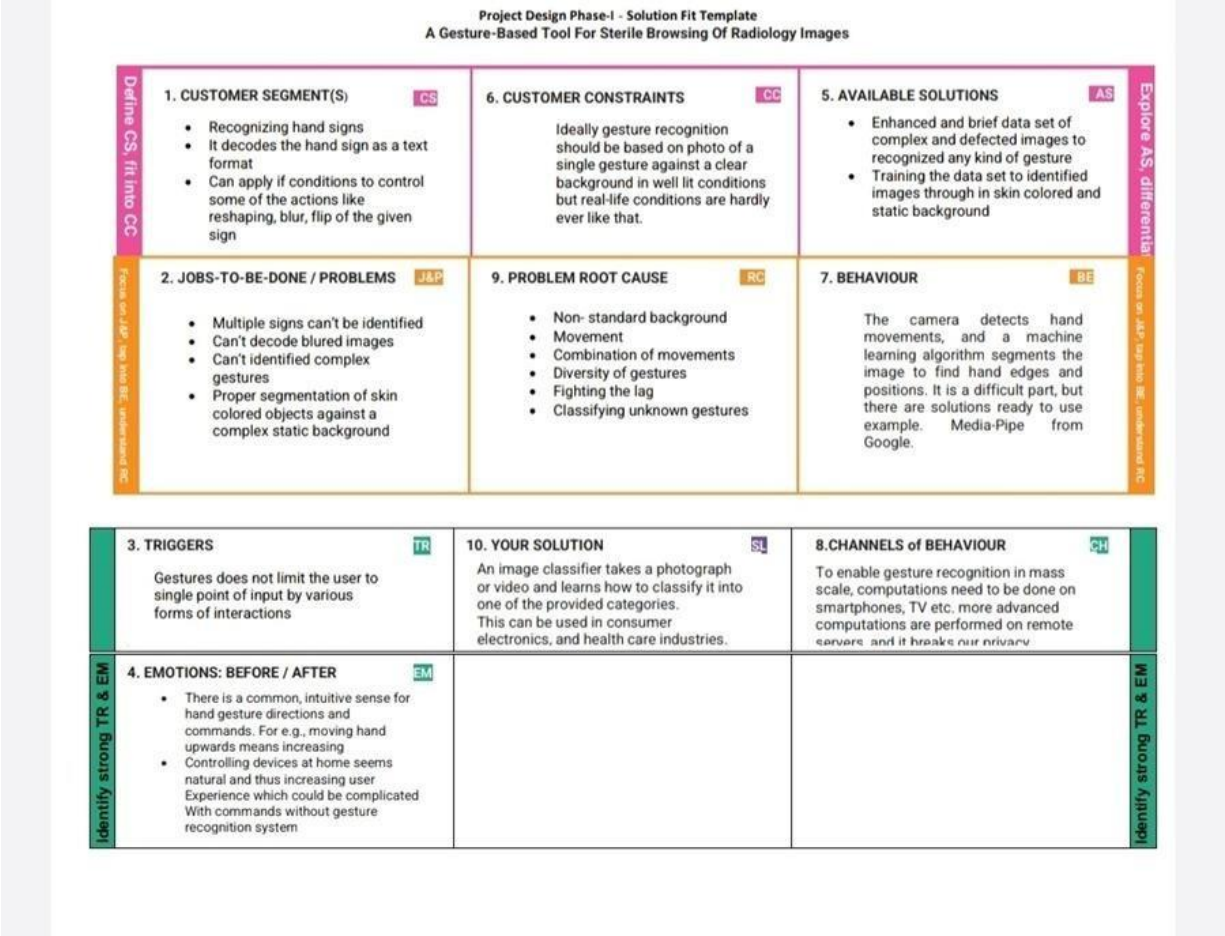


3.3 PROPOSED SOLUTION

This paper presents "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 database. Navigation and other gestures are translated to commands based on their temporal trajectories, through video capture. "Gestix" was tested during a brain biopsyprocedure. In the in vivo experiment, this interface prevented the surgeon's focus shift and change of location while achieving a rapid intuitive reaction and easy interaction. Data fromtwo usability tests provide insights and implications regarding human-computer interaction based on nonverbal conversational modalities.

This paper presents a method to improve the navigation and manipulation of radiological images through a sterile hand gesture recognition interface based on intentional contextual cues. Computer vision algorithms were developed to extract intention and attention cues fromthe surgeon's behavior and combine them with sensory data from a commodity depth camera.The developed interface was tested in a usability experiment to assess the effectiveness of thenew interface. An image navigation and manipulation task wasperformed, and the gesture recognition accuracy, false positives and task completion times were computed to evaluate system performance. Experimental results show that gesture interaction and surgeon behaviour analysis can be used to accurately navigate, manipulate and access MRI images, and therefore this modality could replace the use of keyboard and mice-based interfaces

3.4 PROBLEM FIT SOLUTION:



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

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

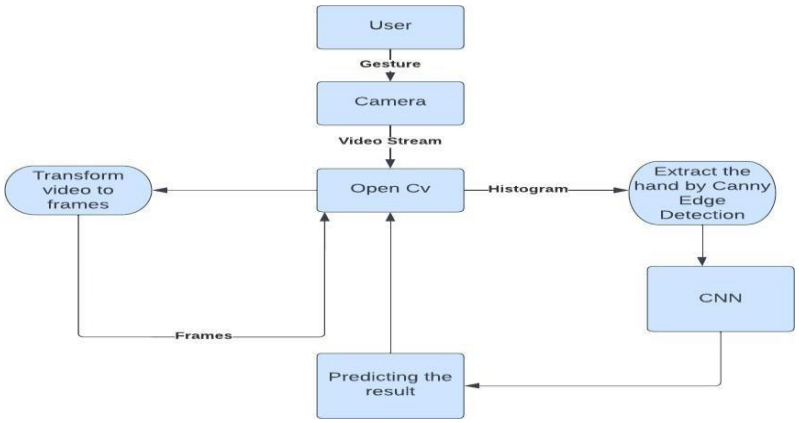
4.2 NON-FUNCTIONAL REQUIREMENTS

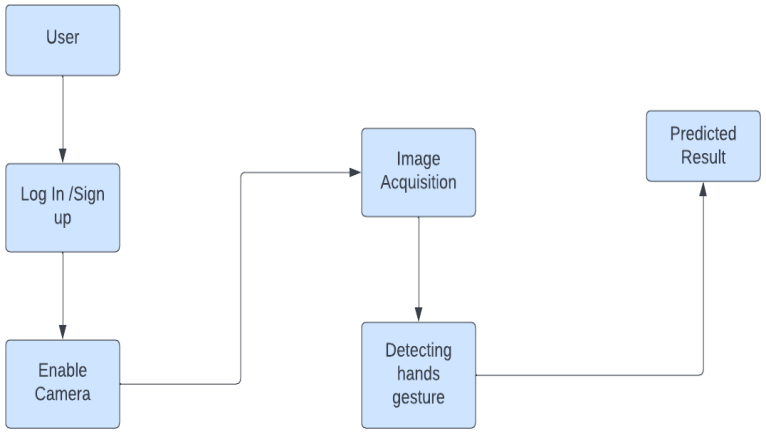
Following are the non-functional requirements of the proposed solution.

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

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM





5.2. Solution & Technical Architecture

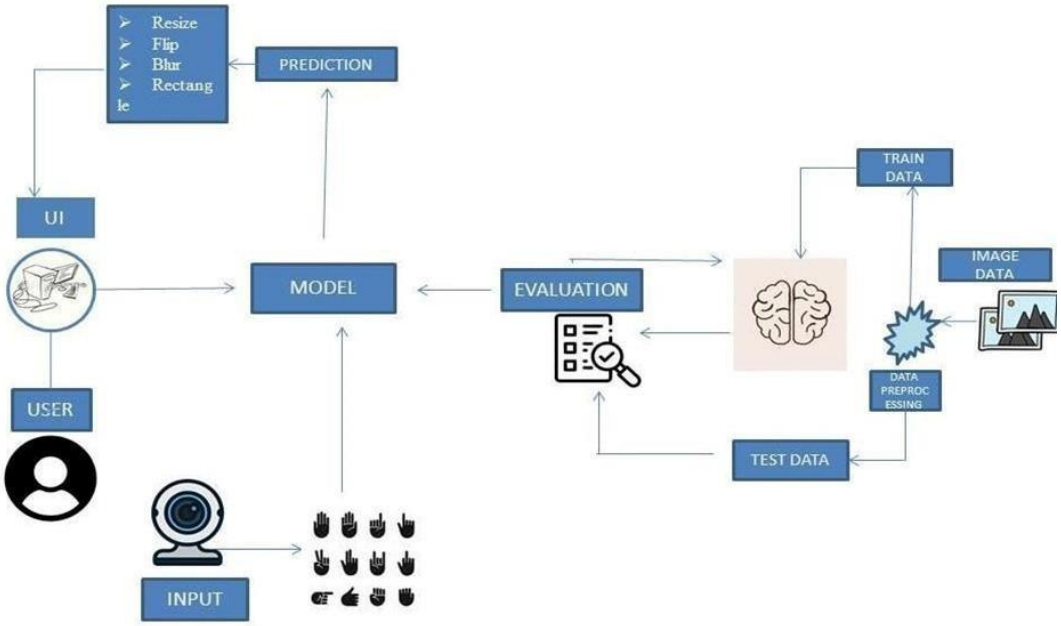


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g.Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular JS / React JS etc.
2.	Application Logic-1	Variety of frameworks, libraries andSupports are required to develop the project	Java / Python

3.	Application Logic-2	Helps to convert the hand signs and hand gestures into the writtenwords to surf on the internet and communicate with computer.	IBM Watson STT service
4.	Application Logic-3	Provides fast, consistent and accurate answers after recognizing the human hand gestures and signs.	IBM Watson Assistant
5.	Database	It can be numerical, categorical or time-series data	MySQL, NoSQL, etc.
5.	Database	It can be numerical, categorical or time-series data	MySQL, NoSQL, etc.
6.	Cloud Database	Enables the user touse host database without buying the additional hardware	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage should be highly flexible, scalable, effective, fast andreliable.	IBM Block Storage or Other StorageService or Local Filesystem
8.	External API-1	Used to access the information in the cloud	IBM Weather API, etc.
9.	External API-2	Used to access the information for datadriven decision making	Aadhar API, etc.
10.	Machine LearningModel	Machine Learning Model deals with various algorithms that are neededfor the implementation	Image Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / CloudLocal Server Configuration: Install the windows version and execute the installer.	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The frame works used in theproject are	Tensor flow, Theano, RNN, pyTorch, Flask
2.	Security Implementations	The security / access controlsare implemented usingfirewalls etc	Firewall and other security related software's.

3.	Scalable Architectu re	the scalability of architecture(3 – tier, Microservices)	Data, models, operate at size, speed, consistency and complexity
4.	Availability	the availability of application (e.g., use of load balancers,distributed servers etc.)	Image and facial recognition, speech recognition and realtime captioning.

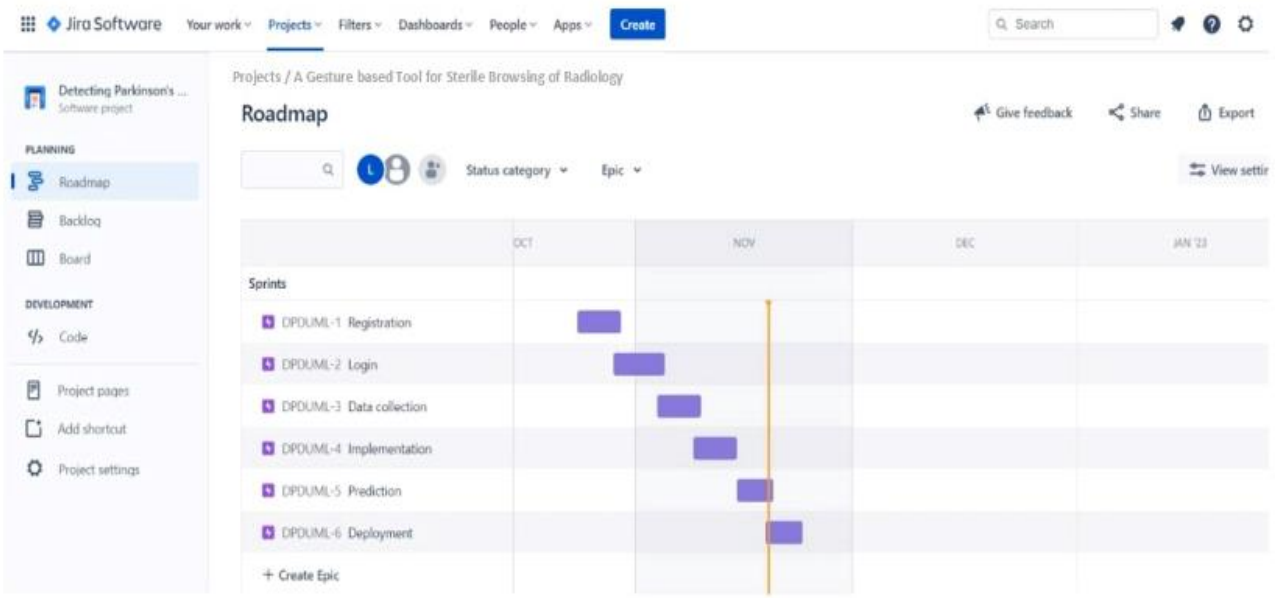
5.PROJECT PLANNING & SCHEDULING

5.1 Sprint Planning , Estimation & Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Registration	USN-1	As a user, I can access the application by entering credential and confirming the password.	2	High
Sprint-1		USN-2	As a user, I will receive the credential to accessthe application	2	High
Sprint-1	Login	USN-3	As a user, I can log into the application byentering the credential	2	Medium
Sprint-2	Dashboard	USN-4	Once logged in , based on data fed during training phase an analysis is made. Once the gesture is displayed the data gets compared tothe one trained and the output is displayed	4	High
Sprint-2	Dataset	USN-5	As a user, I can feed and remove anysort of data	2	High
Sprint-3	Notifications	USN-6	As a user, I can receive notification if the datasetis found similar to the one that is fed.	2	High
Sprint-3		USN-7	As a user, I can check on the desiredoutput.	2	Medium

Sprint-4	Security	USN-8	As a user, I am assured for linking mydatasets securely	4	High
Sprint-4	Customer care	USN-9	As a user, I can access the customer care forany queries and issues regarding the application.	2	Medium

6. Reports from JIRA



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

7.1 Feature 1(login credentials)

```
from tkinter import *
from functools import partial


def validateLogin(username, password):
    print("Hello ", username.get())
    if(password.get()=="key"):
        print("Access Accepted")
```

```
    else:
        print("Access Denied")
    return

tkWindow = Tk()
tkWindow.geometry('400x150')
tkWindow.title('Login Credential')

usernameLabel = Label(tkWindow, text="Credential").grid(row=0, column=0)
username = StringVar()
usernameEntry = Entry(tkWindow, textvariable=username).grid(row=0, column=1)

passwordLabel = Label(tkWindow,text="Key").grid(row=1, column=0)
password = StringVar()
passwordEntry = Entry(tkWindow, textvariable=password, show='*').grid(row=1, column=1)
```

```
validateLogin = partial(validateLogin, username, password)


loginButton = Button(tkWindow, text="Request Access", command=validateLogin).grid(row=4, column=0)


tkWindow.mainloop();
```

7.2 **Feature 2(gesture recognition**)

```
import cv2
import imutils
import numpy as np
from sklearn.metrics import pairwise
from keras.models import load_model
from scipy.misc import imresize
bg = None
def run_avg(image, accumWeight):
    global bg
    if bg is None:
        bg = image.copy().astype("float")
        return
    cv2.accumulateWeighted(image, bg, accumWeight)
def segment(image, threshold=25):
    global bg
    diff = cv2.absdiff(bg.astype("uint8"), image)
    thresholded = cv2.threshold(diff, threshold, 255, cv2.THRESH_BINARY)[1]
    (cnts, _) = cv2.findContours(thresholded.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    if len(cnts) == 0:
        return
    else:
        segmented = max(cnts, key=cv2.contourArea)
        return (thresholded, segmented)
def _load_weights():
    try:
        model = load_model("hand_gesture_recog_model.h5")
        print(model.summary())
        return  model
    except Exception as e:
        return None
def getPredictedClass(model):
    image = cv2.imread('Temp.png')
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    gray_image = imresize(gray_image, [100, 120])
    gray_image = gray_image.reshape(1, 100, 120, 1)
    prediction = model.predict_on_batch(gray_image)
    predicted_class = np.argmax(prediction)
    if predicted_class == 0:
        return "Blank"
    elif predicted_class == 1:
        return "OK"
```

```
    elif predicted_class == 2:
        return "Thumbs Up"
     elif predicted_class == 3:
        return "Thumbs Down"
     elif predicted_class == 4:
        return "Punch"
    elif predicted_class == 5:
        return "High Five"
if__name__== "_main_":
    accumWeight = 0.5
    camera = cv2.VideoCapture(0)
    fps = int(camera.get(cv2.CAP_PROP_FPS))
    top, right, bottom, left = 10, 350, 225, 590
    num_frames = 0
    calibrated = False
    model = _load_weights()
    k = 0
    while (True):
        (grabbed, frame) = camera.read()
        frame = imutils.resize(frame, width=700)
        frame = cv2.flip(frame, 1)
        clone = frame.copy()
        (height, width) = frame.shape[:2]
        roi = frame[top:bottom, right:left]
        gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
        gray = cv2.GaussianBlur(gray, (7, 7), 0)
        if num_frames < 30:
            run_avg(gray, accumWeight)
            if num_frames == 1:
                print("[STATUS] please wait! calibrating...")
            elif num_frames == 29:
                print("[STATUS] calibration successfull...")
        else:
            hand = segment(gray)
            if hand is not None:
                (thresholded, segmented) = hand
                cv2.drawContours(clone, [segmented + (right, top)], -1, (0, 0, 255))
                if k % (fps / 6) == 0:
                    cv2.imwrite('Temp.png', thresholded)
                    predictedClass = getPredictedClass(model)
                    cv2.putText(clone, str(predictedClass), (70, 45), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0,
255), 2)
                cv2.imshow("Thesholded", thresholded)
        k = k + 1
        cv2.rectangle(clone, (left, top), (right, bottom), (0, 255, 0), 2)
        num_frames += 1
        cv2.imshow("Video Feed", clone)
        keypress = cv2.waitKey(1) & 0xFF
        if keypress == ord("q"):
```

```
        break
    camera.release()
    cv2.destroyAllWindows()
```

7.3 Database Schema (if Applicable)

8.TESTING

8.1 Test Cases

Test case 1:

Detectiong hand gesture for 5 (stop_)

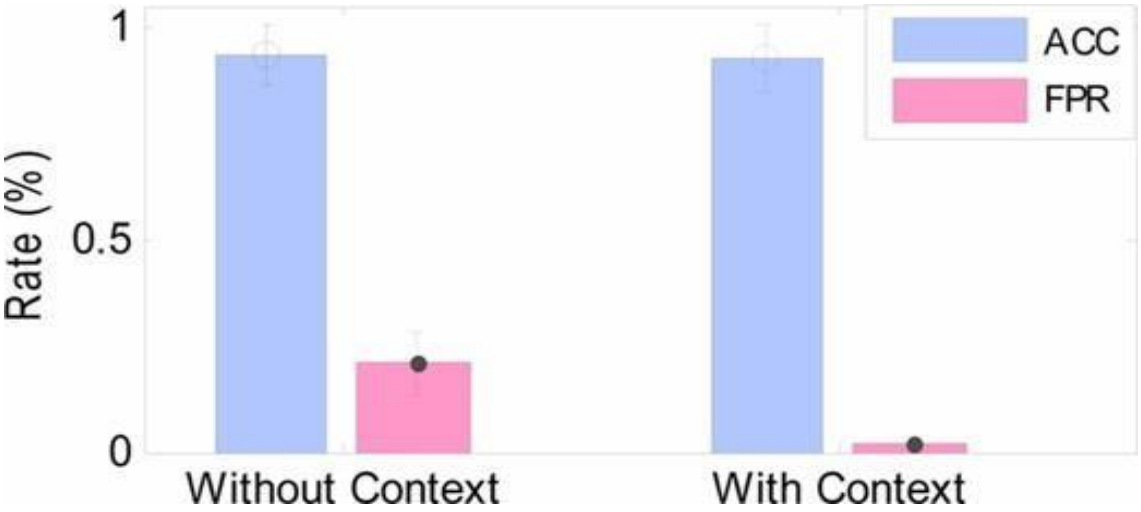
```
DATASET_PATH = 'C:\Users\MERESHIYA\GAYATHRI\JENI\KAVYADataset\data\five\'

gesture_path = os.path.join(DATASET_PATH, '*')
print(gesture_path)
import glob
gesture_path = glob.glob(gesture_path)

rand_index = random.randint(0, len(gesture_path))
print(len(gesture_path))
image = cv2.imread(gesture_path[rand_index])
image = imresize(image,[100, 120])
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
```

Test case 2:

Detecting thumbs_down:

```
import cv2
image = cv2.imread('thumbsdown44.jpg')
image = imresize(image,[100, 120])
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
```

8.2 USER ACCEPTANCE TESTING



Comparison of gesture recognition with and without context.The sterile radiology image recognition is done easy to make job easier. The mean gesture recognition accuracy (ACC) across all users and trials in the experiment testing in-task performance remains approximately the same with and without context (92.58% and 93.6%, respectively), and the false positive rate (FPR) drops from 20.76% to 2.33% upon integrating context. This figure is only reproduced in colour in the online version.

Mean task completion times of the first two and last two trials in experiment 3

	Trial No	Mean task completion time for trial (s)	Mean task completion time (s)
Initial	1	68.00	75.05
	2	82.11	
Final	9	49.91	47.14
	10	44.38	

During the ‘non-intentional’ phase of each trial, segmented gestures (false positives) were recognized. In the system without contextual cues, a FPR of 20.76% was obtained, whereas the FPR was reduced to 2.33% when contextual cues were integrated. One-way analysis of variance indicated that the mean task completion time of the first two trials (75.05 s) was significantly ($p<0.05$) longer than the mean task completion time of the last two trials (47.14 s).

The gestures performed by one typical user were analyzed to determine the maximum and minimum ranges of each gesture as functions of maximum grasp range (MGR) (defined as the radius of the arcs from the shoulder pivots19). The increase brightness gesture (table 1, gesture g) required the most reach (100% MGR), and zoom in (table 1, gesture i) required the least reach (43.79% MGR).

Analysis of questionnaire data (maximum score of 5) showed that users rated the lexicon of gestures designed with surgeons as easy to learn (4.40±0.68) and remember (4.05±0.94) and moderately easy to perform (3.60±0.88).

9. RESULTS

Final findings (Output) of the project along with screenshots.Through this project we found that we can maintain the sterility of an operation theater, etc by using hand 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

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.

12. FUTURE SCOPE

The system could also be made smart to be trained for only one or two gestures rather than alland then made ready for testing. This will require only a few changes in the current interface code, which were not performed due to the shortage of time.

One time training constraint for real time system can be removed if the algorithm is made efficient to work with all skin types and light conditions which seems impossible by now altogether. Framing with COG (Centre of gravity) to control orientation factor could make this system more perfect for real application. The system‘s speed for preprocessing could beimproved if the code is developed in VC/VC.Net. Tracking of both hands can be added to

increase the set of commands. Voice commands can also be added to further increase thefunctionality.

13. APPENDIX (SOURCE CODE):

```
import numpy as np
import os
from scipy.misc.pilutil import imread, imresize
import datetime
from skimage import io
import os
import random
import matplotlib.pyplot as plt
% matplotlib inline
import glob
import glob
import numpy as np
import cv2
np.random.seed(30)
import random as rn
rn.seed(30)
from keras import backend as K
import tensorflow as tf
tf.set_random_seed(30)
def plot_image(images, captions=None, cmap=None ):
    f, axes = plt.subplots(1, len(images), sharey=True)
    f.set_figwidth(15)
    for ax,image in zip(axes, images):
        ax.imshow(image, cmap)
import cv2

cat = cv2.imread('C:\Users\MERESHIYA\Dataset\data\five\hand1(984).jpg')
plt.imshow(cv2.cvtColor(cat, cv2.COLOR_BGR2RGB))
```

```
DATASET_PATH = 'C:\Users\MERESHIYA\Dataset\data\five\'

gesture_path = os.path.join(DATASET_PATH, '*')
print(gesture_path)
import glob
gesture_path = glob.glob(gesture_path)

rand_index = random.randint(0, len(gesture_path))
print(len(gesture_path))
image = cv2.imread(gesture_path[rand_index])
image = imresize(image,[100, 120])
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
DATASET_PATH = 'C:\Users\MERESHIYA\Datase\data'

dataset_path = os.path.join(DATASET_PATH, '*')
import glob
dataset_path = glob.glob(dataset_path)
dataset_path
loaded_images = []

list_of_gestures = ['blank', 'ok', 'thumbsup', 'thumbsdown', 'fist', 'five']

for path in range(0, len(dataset_path)):
    dataset_path = "C:\Users\MERESHIYA\Dataset\data\" + str(list_of_gestures[path])
    print(dataset_path)
    gesture_path = os.path.join(dataset_path, '*')
    import glob
    gest_path = glob.glob(gesture_path)
    print(len(gest_path))
    k = 0
    for i in range(0, len(gest_path)):
        if k < 1600:
            image = cv2.imread(gest_path[i])
            gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            gray_image = imresize(gray_image,[100, 120])
            loaded_images.append(gray_image)
        k=k+1

len(loaded_images)
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, BatchNormalization
from keras.layers import Activation, Dropout
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K
from keras.optimizers import Adam

# model
model = Sequential()
```

```
# first conv layer
# input shape = (img_rows, img_cols, 1)
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(100,120, 1)))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# second conv layer
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# flatten and put a fully connected layer
model.add(Flatten())
model.add(Dense(128, activation='relu')) # fully connected
model.add(Dropout(0.5))

# softmax layer
model.add(Dense(6, activation='softmax'))

# model summary
optimiser = Adam() #write your optimizer
model.compile(optimizer=optimiser, loss='categorical_crossentropy',
metrics=['categorical_accuracy'])
model.summary()
outputVectors = []
for i in range(1, 1601):
    outputVectors.append([1, 0, 0, 0, 0, 0])

for i in range(1, 1601):
    outputVectors.append([0, 1, 0, 0, 0, 0])

for i in range(1, 1601):
    outputVectors.append([0, 0, 1, 0, 0, 0])

for i in range(1, 1601):
    outputVectors.append([0, 0, 0, 1, 0, 0])

for i in range(1, 1601):
    outputVectors.append([0, 0, 0, 0, 1, 0])

for i in range(1, 1601):
    outputVectors.append([0, 0, 0, 0, 0, 1])

len(outputVectors)
import cv2
image = cv2.imread('thumbsdown44.jpg')
image = imresize(image,[100, 120])
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
```

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray_image = imresize(gray_image,[100, 120])

gray_image = gray_image.reshape(1, 100, 120, 1)
gray_image.shape
model.save("trained_number_gesture.h5")
from keras.models import load_model
model = load_model("hand_gesture_recog_model.h5")
model.summary()
pred_idx = np.argmax(model.predict_on_batch(gray_image), axis=1)
pred_idx
```

GITHUB & PROJECT DEMO LINK:

https://github.com/IBM-EPBL/IBM-Project-12724-1659459684

https://drive.google.com/file/d/1jBmZKeNU_SiVCqr9D3UEPRv2j_C4dPtn/view?usp=drivesdk