A GESTURE-BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

Team ID: PNT2022TMID00221

Submitted by:

AARTHI A
AKSHAYA E L
ARTHI D
DEVADHARSHINI M

A Gesture Based Tool for Sterile Browsing of Radiology Images

1.INTRODUCTION

1.1 PROJECT 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 **EXISTING PROBLEM:**

In the twenty-first century, there has been a close interaction between social life and information technology. Future consumer electronics items will feature increasingly complicated interfaces with a wide range of functionalities. How to create a practical human machine. The user interface of every consumer electronics product is now a crucial concern. The most popular formof interaction is still using a mouse, keyboard, or joystick, which are all classic examples of electronic input devices. It does not imply, however, that these gadgets are the most practical and natural input methods for the majority of users. Gestures have been an important form of human contact and communication since ancient times. Before the development of language, people could simply convey their ideas through gestures. Many individuals still use gestures in everyday life, and deaf people in particular find that gestures are the most natural and important form of communication. The gesture control method has recently emerged as a new trend in the development of many human-based electronics products. Development of a real-time hand gesture identification system based on an adaptive colour HSV model and motion history image is the goal of this research (MHI). The influences of lighting, environment, and camera may be significantly minimised by using an adaptive skin colour model, and the robustness of hand motion detection might be significantly increased.

2.2 REFERENCE:

First, there are a number of approaches utilized in the vision-based approach for hand detection, training gestures, background subtraction, and finger tip detection, which are reviewed as follows: Scale invariant feature transform [1][3] based hand detection as well as the feature-based hand detection methods utilized by Viola and Jones detector have been implemented. These algorithms deliver very accurate results but are more prone to background noise. The second method is picture segmentation, which instead of using the RGB color model to determine the color of human skin, employs the HSV color space model. This approach providesbetter results for background separation and region boundaries but is unable to identify objects with similar-colored backgrounds that have different skin tones [1]. The third method incorporates data from the same category of items using learning-based gesture recognition in the Adaptive Boosting algorithm. By integrating all weak classifiers into one strong classifier, it trains the network. The best weak classifier is chosen from a set of both positive and negative image examples by the AdaBoost learning algorithm [2]. This technique delivers results more quickly and accurately, although occasionally the network training process takes longer. Anothermethod is to look for convex hulls. For palm detection, there are so many algorithms available. Some of the existing algorithms that are used in our suggested technique will be detailed in this section.

The Divide and Conquer algorithm, the Quick Hull algorithm, the Jarvis March or Gift-wrapping algorithm, and the Graham's Scan algorithm. Any given set of points' convex hull can be calculated using Graham Scan. There is no requirement for the user to touch or carry a peripheraldevice in order to deploy the system for real-time hand tracking and basic gesture detection. By comparing the results, we can draw the conclusion that using just one detection strategy is insufficient because several techniques can be used to address various detection and recognition issues. AdaBoost, support vector machine technique, hidden Markov model, and principal component analysis are some of the machine learning techniques that are now accessible for training classifiers. The boundary of the hand region may also have a different convex hull and contour detection. Using the adaptive boosting algorithm for hand detection and the Haar classifier algorithm to train the classifier, we will develop the system based on all of these techniques. Convex hull technique is used to form a contour around the palm and finger tip

detection in this case, along with the HSV colour model for background reduction and noise removal.

While there are several image processing software programme available, OpenCV (Open Source for Computer Vision) is particularly well-liked for real-time image processing tasks like gesture and object recognition. The ability to simply combine the code with hardware is the main benefit. We put the suggested approach into practise using the Linux-based OpenCV library. Six 2D convolution layers make up the network, and a max-pooling operator comes after each layer. Fig displays the volumes at each layer, the convolution kernel sizes, and the pooling operators. A fully linked network with nine layers receives the output of the sixth convolution layer as input.

Except for the last output layer, which includes nine neurons—one for each of the nine hand gestures—each layer comprises 512 hidden neurons. In the output layer, a sigmoid activation function is applied. The remaining eight levels employ the Tanh activation function. The acquisition of a big dataset for each subject in the context of this article would be time-consuming and impractical when taking into account real-life applications, as a user would frequently not put up with hours of data recording for each training.

The initial stage for any system is to gather the information required to carry out a certain operation. Input data for hand posture and gesture recognition systems is collected using a variety of technologies. This project can meet the requirements of the user's need by tracking themotion or movement of the hand. At the Washington Hospital Center in Washington, DC, the gesture interface's functionality was evaluated. A hand gesture system's applicability was learnedfrom watching two operations in the hospital's neurosurgery department. This is the first instancethat we are aware of where a hand gesture recognition system has been effectively used in a "in vivo" neurosurgical biopsy. Batch Normalization [4] is used to solve this overfitting problem in more depth.

[5] A sterile human—machine interface is of supreme importance because it is the means by which the surgeon controls medical information avoiding contamination of the patient, the OR and the surgeon.

[6]Hand gestures are a fundamental type of nonverbal communication. Psychological research has shown that before they can speak, young children communicate through gestures. When people talk to one another about an object, manipulation is a common form of gesticulation. There are several compelling reasons to switch from the current interface technology (such as the keyboard, mouse, and joystick) to more natural interfaces, including naturalness of expression, unrestricted interaction, intuitiveness, and high sterility. At the Washington Hospital Center in Washington, DC, the gesture interface's functionality was evaluated. A hand gesture system's applicability was learned from watching two operations in the hospital's neurosurgery department.This is the first instance that we are aware of where a hand gesture recognition system has been effectively used in a "in vivo" neurosurgical biopsy. A sterile human-machine interface is crucial because it allows the surgeon to control medical information without contaminating the patient, the operating room, or themselves.

[7]More people are getting cancer, and patients are older. Often, invasive surgery, or the removal of the lesion, cannot cure patients, especially those who are elderly and have several diseases.

The radiologist percutaneously inserts energy applicators with high precision into the tumour location with the aid of imaging tools like CT scanners. However, because the majority of CT systems were created as diagnostic tools, they lack the support and interaction paradigms necessary for the interventional workflow. The radiologist must place the X-Ray tube and detector at the specified angle for radiography when using flat panel CTs to capture images that show the positions of the instrument and risk structures. Due to sterility difficulties caused by theusage of touchscreens or haptic buttons wrapped in a sterile drape, observations showed that the current workflow is not appropriate.

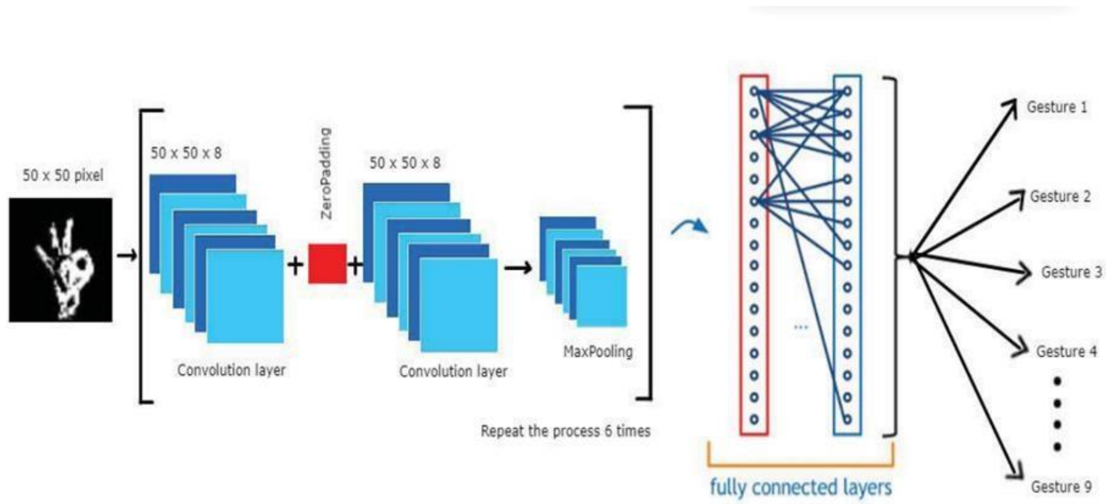


Fig :The network consists of 6 convolutional + max pooling layers, output of the 6th layer is given as input to a fully connected neural network with 9 hidden layers. Each hidden layer has 512 neurons, except the output layer which has 9 neurons, one each for each hand gesture.

[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 Human-Computer 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,

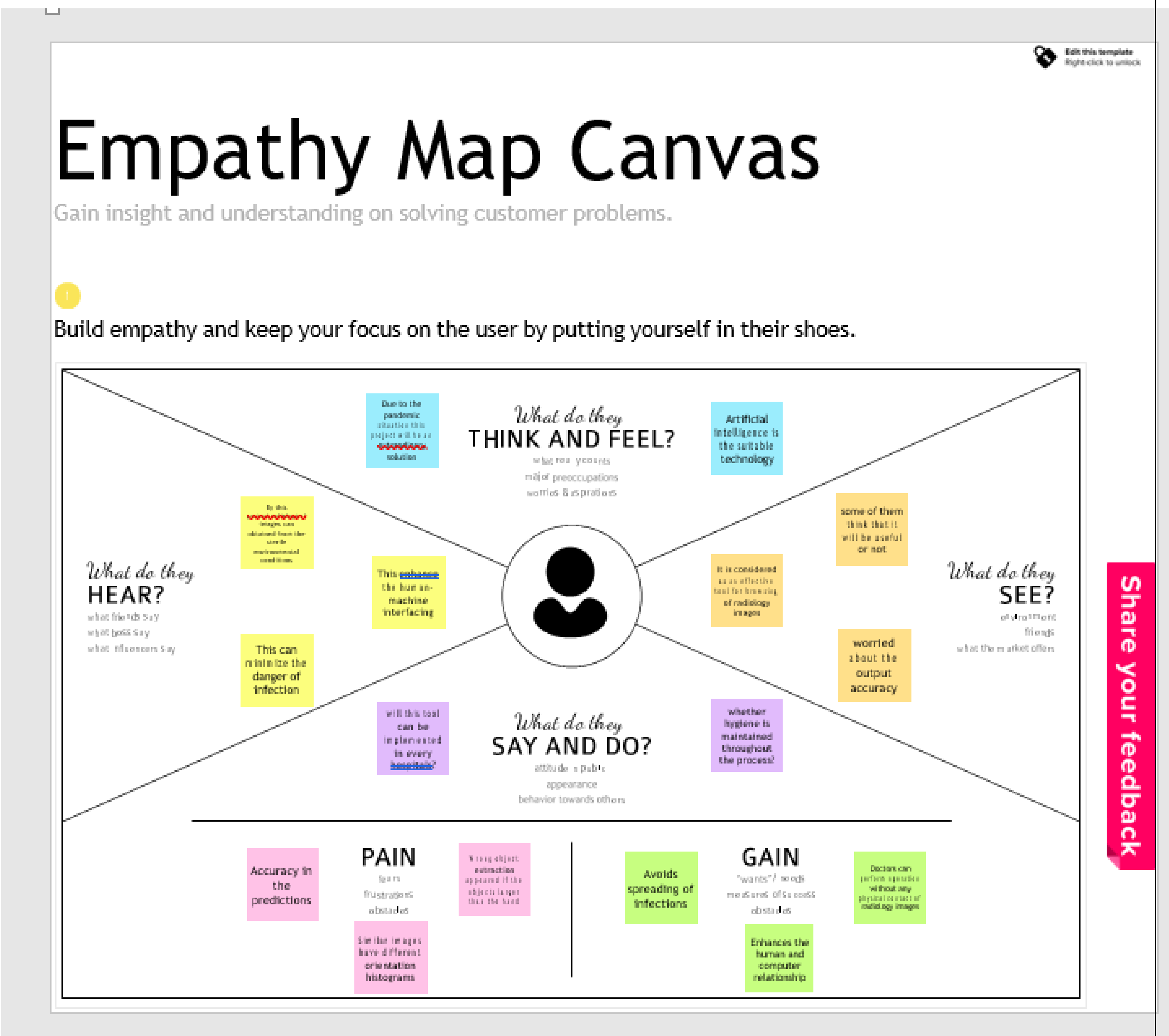
[7]Professor. Benjamin Fritsch,

2.3 PROBLEM STATEMENT DEFINITION:

Physicians face a difficult problem when interacting with interventional imaging devices in a sterile setting. Because of sterility requirements and workspace limitations, direct physician-machine interaction during an intervention is very restricted. During computed tomography (CT)-based interventions, we provide a way of gesture-controlled projection display that enables a direct and natural physician-machine interface. Therefore, a radiation barrier in front of the doctor is projected with a graphical user interface. Using a jump motion controller, hand motions made in front of this display are recorded and categorized.

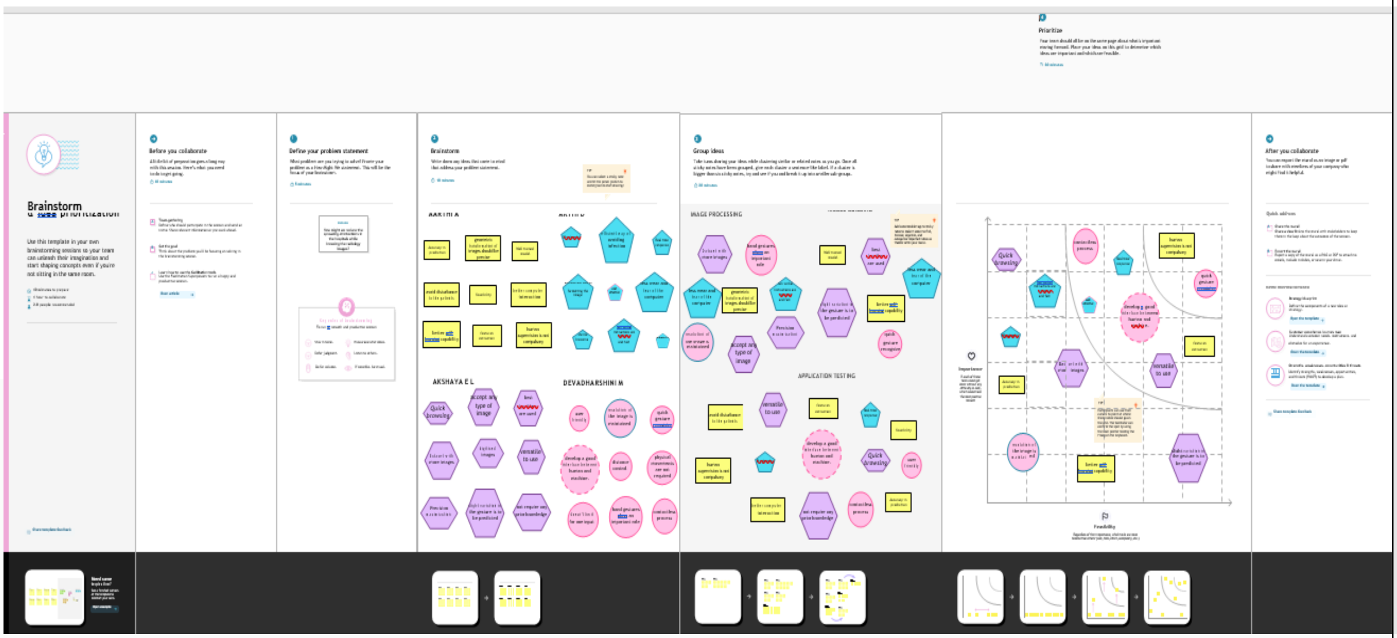
3 IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS:



ate

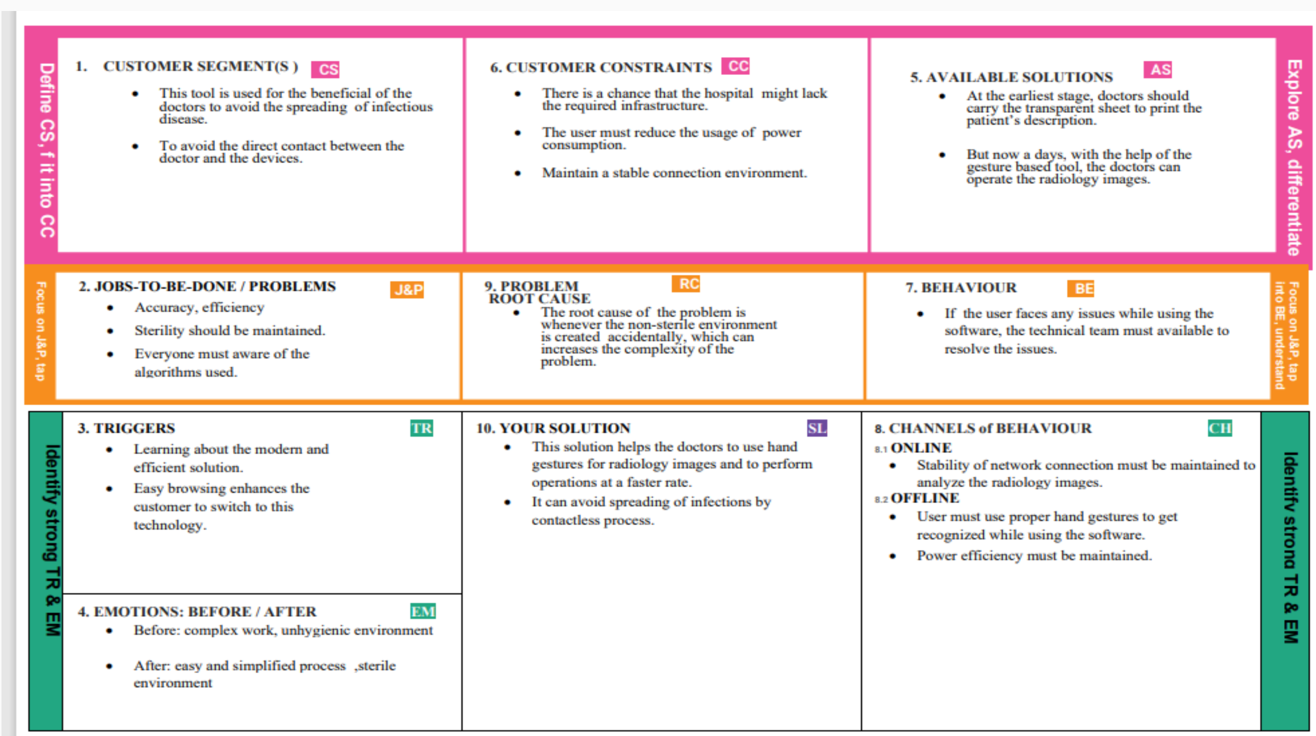
3.2 IDEATION AND BRAINSTORMING :



3.3 PROPOSED SOLUTION:

To avoid the infection spreading by touching the keyboard, mouse etc.. during the browsing of radiology images and to maintain the sterility, a hand gesture based tool for browsing of radiology images is made as a solution for this problem.

3.4 PROBLEM SOLUTION FIT:



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	User Registration	Registration is done while using the application.
FR-2	User Confirmation	Confirmation is done by many methods like sending OTP to the user Mail or phone number.
FR-3	Authentication	Authentication is done by the user when the user is using the application to scan the radiology images.
FR-4	Network	Ethernet or Wi-Fi should be provide throughout to improve the performance of the application to access the resources.
FR-5	Sterile environment	Sterile environment without any interferences is required.
FR-6	Clarity of Images	The test and trained images must be visibility and easily recognizable.

4.2 NON-FUNCTIONAL REQUIREMENTS:

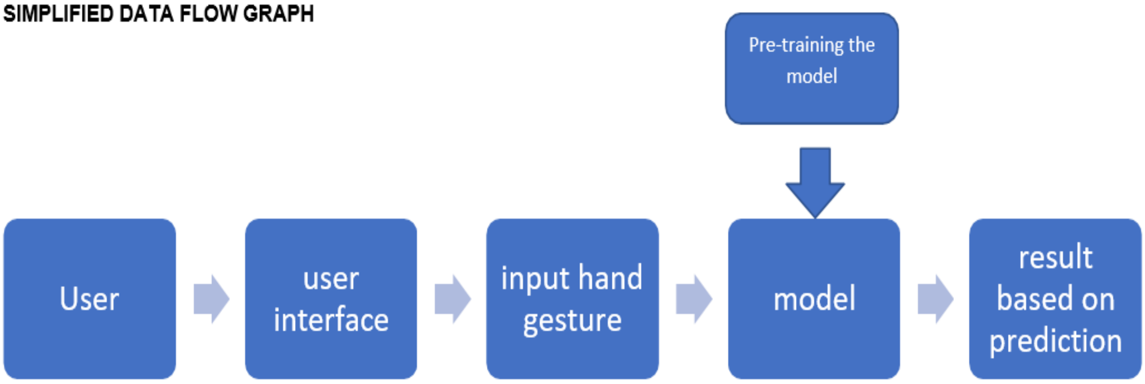
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Images can be accessed with no direct contact and communicate with the gesture itself
NFR-2	Security	The only form to maintain the security in the web Application is by Encrypting the data which will not be hacked.
NFR-3	Reliability	Easy to use, which will recognize the gesture images without the contact.
NFR-4	Performance	The recognition of the gesture images is normally speed and accuracy is maintained.
NFR-5	Availability	It is available only to the people who has authority to access the resources or application.
NFR-6	Scalability	Scalability is commonly high where the multiple users can use the application from anywhere at any time.

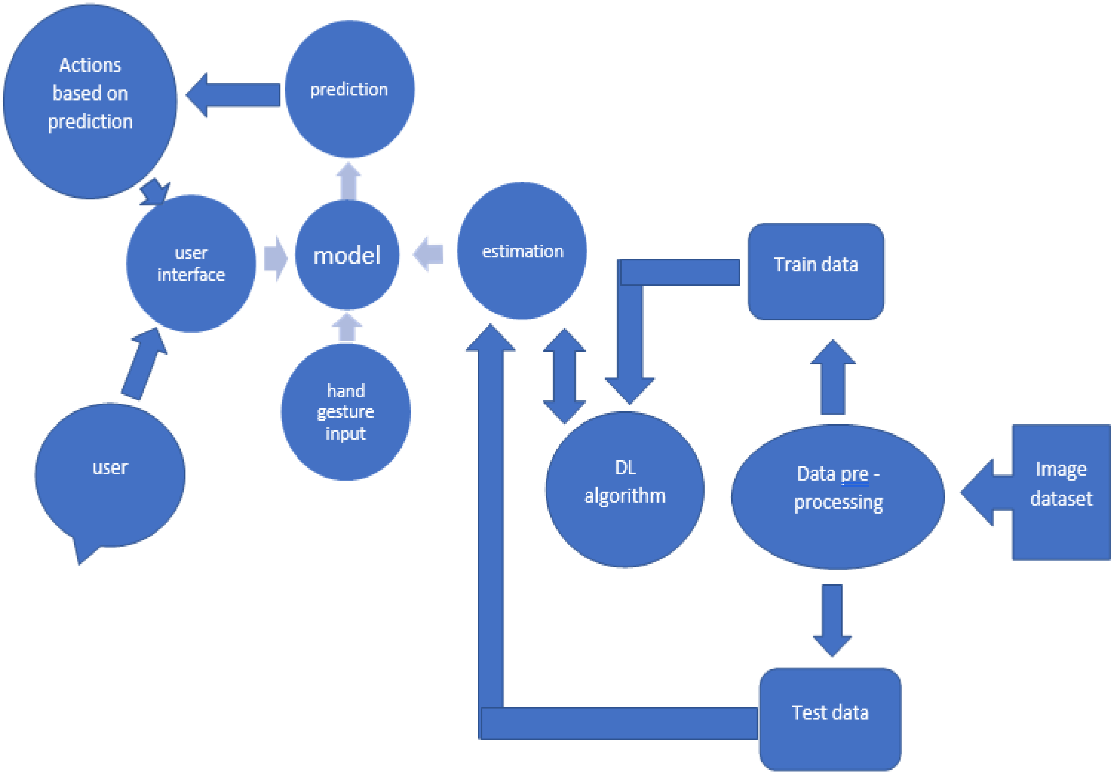
5 PROJECT DESIGN :

5.1 DATA FLOW DIAGRAM

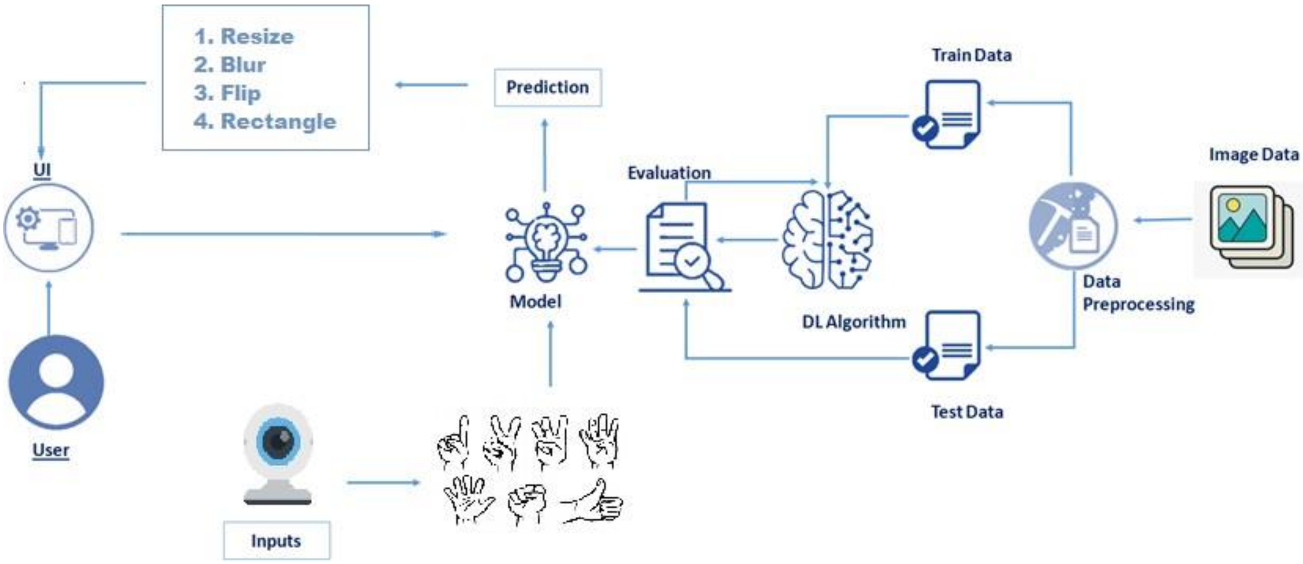
SIMPLIFIED DATA FLOW GRAPH



INDUSTRIAL STANDARD DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE:



5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Webpage(application)	USN-1	Entering into web page application	I can access my web page	High	Sprint-1
	Upload	USN-2	As a user, I can upload my radiology image which I want to blur,rotate,zoom etc...	I can upload the image from my device	High	Sprint-1
	User interface	USN-3	As a user, I can show my hand gesture in front of the camera by turning the camera ON	I can show hand gesture	High	Sprint-2
	To know more	USN-4	As a user to know more about the application, I can click 'more'.	I can get more information	low	Sprint-3
	Model building	USN-5	Building the model ,training the and testing the model using DL ALGORITHM	I can train and test the model efficiently	High	Sprint-2
	Hand gestures	USN-6	Based on the pre-trained model,if it predicts 1 then the image is blurred	I can get the output as blurred	High	Sprint-2
		USN-7	Based on the pre-trained model, if it predicts 2 then the image is resized	I can get the output as resized	High	Sprint-2
		USN-8	Based on the pre-trained model,if it predicts 3 then the image is rotated	I can get the output as rotated	High	Sprint-2
	Procedure	USN-9	The procedure is simple, user friendly and common to every user	I can access the features as a common man	Medium	Sprint-3
	Accurate feature extraction	USN-10	The best and correct output for the given hand gesture is required	I can get required action on my image	High	Sprint-3
	Flask interfacing	USN-11	To interface the python and webpages for application development	To create a virtual environment	High	Sprint-4

6.PROJECT PLANNING AND SCHEDULING:

6.1 SPRINT PLANNING AND ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data collection	USN-1	Download the dataset	10	High	Arthi D
Sprint-1		USN-2	Image preprocessing	10	High	Akshaya E L Devadharshini M
Sprint-1		USN-3	Import and configure the image data generator library and class	10	High	Aarthi A Arthi D
Sprint-1		USN-4	Apply image data generator functionality to Trainset and Testset	10	High	Aarthi A Akshaya E L
Sprint-2	Model building	USN-5	Building the model ,training and testing the model using DL ALGORITHM	10	High	Aarthi A Devadharshini M
Sprint-2		USN-6	Adding CNN layers and Dense layers	10	High	Akshaya E L Arthi D
Sprint-2		USN-7	Configure the learning process	10	Medium	Aarthi A
Sprint-3	Webpage(Application)	USN-8	Create web application using HTML,CSS,JS	10	High	Aarthi A Akshaya E L Arthi D Devadharshini M
Sprint-3	Hand gestures	USN-9	Based on the pre-trained model, if it predicts 1 then the image is resized	10	High	Akshaya E L
Sprint-3		USN-10	Based on the pre-trained model, if it predicts 2 then the image is rotated	10	High	Arthi D
Sprint-3		USN-11	Based on the pre-trained model ,if it predicts 3 then the image is blurred	10	High	Devadharshini M
Sprint-4	Train and Test the model	USN-12	Train and Test the model and its overall performance	10	High	Aarthi A Akshaya E L Arthi D Devadharshini M

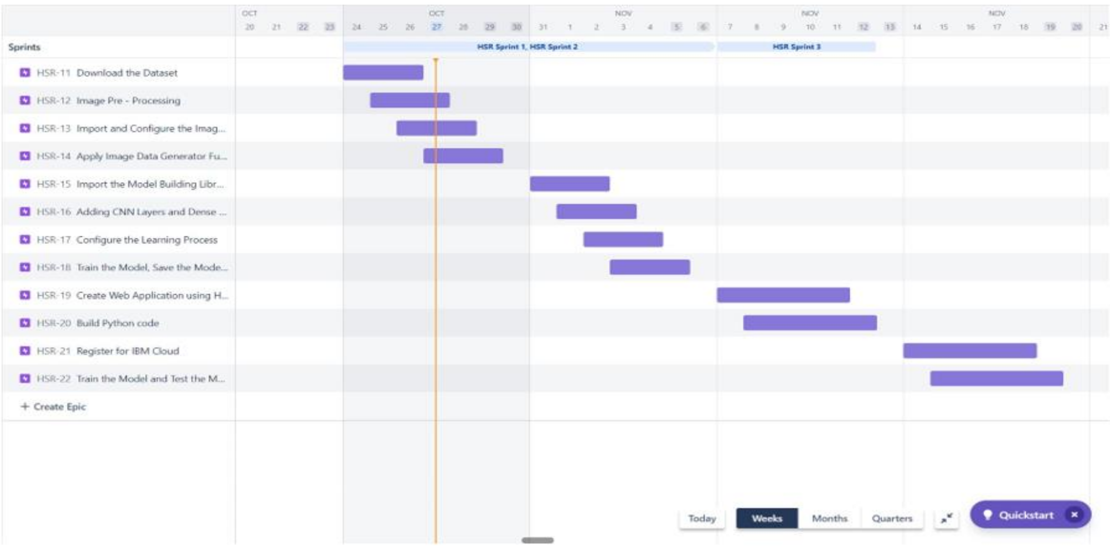
6.2 SPRINT DELIVERY SCHEDULE:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	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:



7.CODING AND SOLUTIONING:

7.1 Feature 1

IMAGE PREPROCESSING:

Import the ImagesDataGenerator Library

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

Configure ImageDataGenerator Functionality To Trainset And Testset

```
Image Data Agumentation

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

Apply ImageDataGenerator Functionality To Trainset And Testset

Loading our data and performing data agumentation

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

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

Model training

Importing The Model Building Libraries

Importing Neccessary Libraries

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

Initializing The Model

```
model=Sequential()
```

Adding CNN Layers

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

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

Adding Dense Layers

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

```
classifier.summary()#summary of our model
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 62, 62, 32)        320
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten_3 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_6 (Dense)              (None, 128)               802944
_________________________________________________________________
dense_7 (Dense)              (None, 6)                 774
=================================================================
Total params: 813,286
Trainable params: 813,286
Non-trainable params: 0
_________________________________________________________________
```

Configure The Learning Process

Training Model

Fitting the model

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

Save the Model

Saving our model

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

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

Test the Model

Predicting our results

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

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

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

Compiling the model

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

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

```
'1'
```

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	Stat
HomePage_TC_OO1	UI	Home Page	Verify user is able to see the home page for our gesture based tool for sterile browsing of radiology images	-	1.Enter URL and click go 2. Check whether home page can be seen	http://127.0.0.1:5000/home	The homepage should be displayed	Working as expected	Pas
HomePage_TC_OO2	UI	Home Page	Verify whether the navigation bar is displayed	-	1.Enter URL and click go 2. Check whether the navigation bar is displayed in the homepage	127.0.0.1:5000/home	The navigation bar should be displayed in the homepage	Working as expected	Pas
IntroPage_TC_001	UI	Intoduction Page	Verify user is able to see the introduction page for our gesture based tool for sterile browsing of radiology images	-	1.Enter URL and click go 2. Check whether intoduction page can be seen	http://127.0.0.1:5000/intro	The Introduction should be displayed	Working as expected	Pas
LaunchPage_TC_001	UI	Launch Page	Verify whether ther user is able to access the launch page	-	1.Enter URL and click go 2. Check whether launch page can be seen	127.0.0.1:5000/index6	The launch page should be displayed	Working as expected	Pas
LaunchPage_TC_002	UI	Launch Page	Verify if the user could see the utility for uploading the rediology image	-	1.Enter URL and click go 2. Check whether the launch page upload image utility can be seen	127.0.0.1:5000/index6	The utility to upload image should be displayed	Working as expected	Pas
LaunchPage_TC_003	Functional	Launch Page	Verify if the user can upload the image from their local system	-	1.Enter URL 2.Click on upload image utility 3. Choose the image that needs to be	http://127.0.0.1:5000/index6	The name of the image uploaded should be displayed	Working as expected	Pas
LaunchPage_TC_004	Functional	Launch Page	Verify if the user can submit	-	1.Enter URL 2.Click on upload image utility 3. Choose the image thet needs to be manipulated 4. Click Submit	http://127.0.0.1:5000/index6 Upload the image to be manipulated	The web-camera starts	Working as expected	Pas
LaunchPage_TC_005	Functional	Launch Page	Verify if the image uploaded is accessible in the backend	-	1.Enter URL 2.Click on upload image utility 3. Choose the image thet needs to be manipulated 4. Click Submit	http://127.0.0.1:5000/index6 Upload the image to be manipulated Click Enter	The image is stored in a folder in the backend	Working as expected	Pas
LaunchPage_TC_006	Functional	Launch Page	Verify if the web cam starts	-	1.Enter URL 2.Click on upload image utility 3. Choose the image thet needs to be manipulated 4. Click Submit	http://127.0.0.1:5000/index6 Upload the image to be manipulated Click Enter	The webcam starts	Working as expected	Pas
LaunchPage_TC_011	Functional	Launch Page	Verify if the gesture 3 shown in camera is recogized	-	1.Enter URL 2.Click on upload image utility 3. Choose the image thet needs to be manipulated 4. Click Submit 5. Show gesture in the webcam	http://127.0.0.1:5000/index6 Upload the image to be manipulated Click Enter Show gesture 3 in the webcam	The uploaded image should be blurred and should be shown	Working as expected	P
LaunchPage_TC_012	Functional	Launch Page	Verify if the gesture 4 shown in camera is recogized	-	1.Enter URL 2.Click on upload image utility 3. Choose the image thet needs to be manipulated 4. Click Submit 5. Show gesture in the webcam	http://127.0.0.1:5000/index6 Upload the image to be manipulated Click Enter Show gesture 4 in the webcam	The uploaded image should be zoomed and should be shown	Working as expected	P
					1.Enter URL	http://127.0.0.1:5000/index6			

8.2.USER ACCEPTANCE TESTING:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the gesture-based tool for sterile browsing of radiology images project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

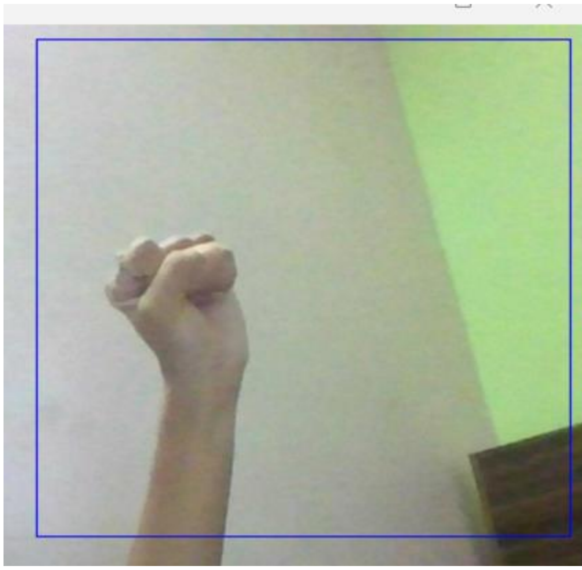
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	3	2	1	0	6
Duplicate	1	0	0	0	1
External	2	3	0	2	7
Fixed	3	2	1	0	6
Not Reproduced	0	0	1	1	2
Skipped	0	0	0	1	1
Won't Fix	0	2	1	0	3
Totals	9	9	4	4	26

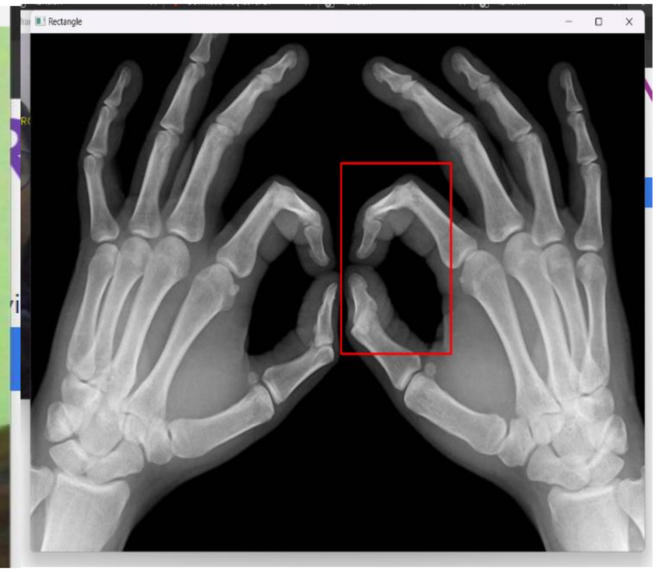
3. Test Case Analysis

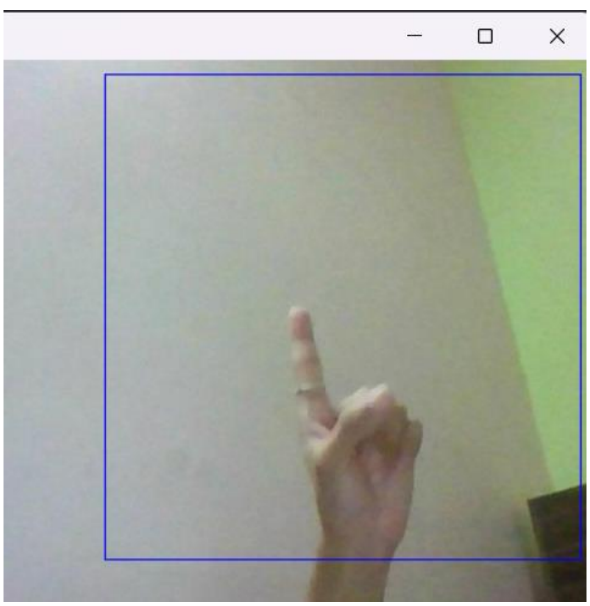
This report shows the number of test cases that have passed, failed, and untested

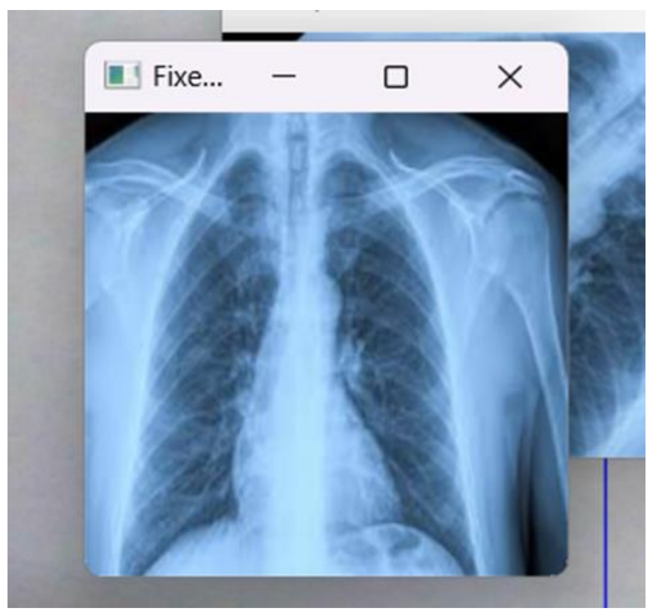
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	16	0	0	16
Outsource Shipping	3	0	0	3
Exception Reporting	2	0	0	2
Final Report Output	7	0	0	7
Version Control	2	0	0	2

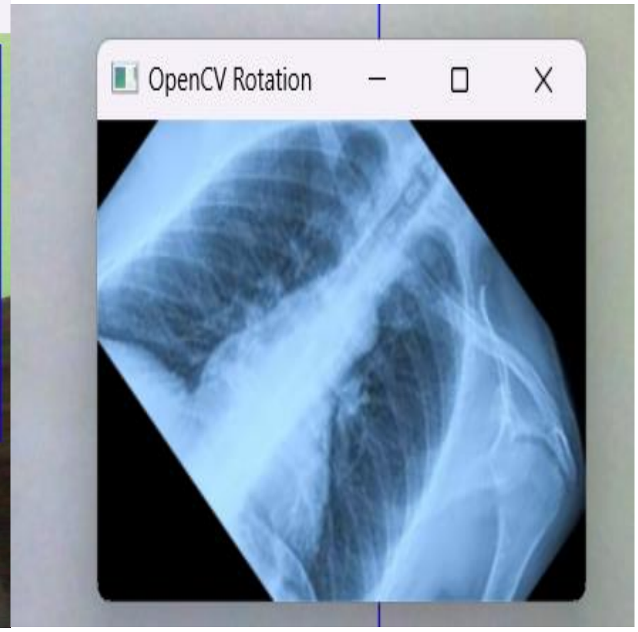
Final findings (Output) of the project along with screenshots.

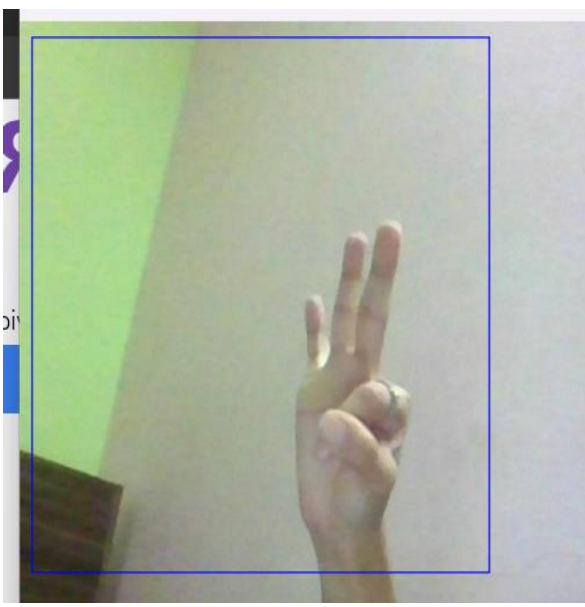


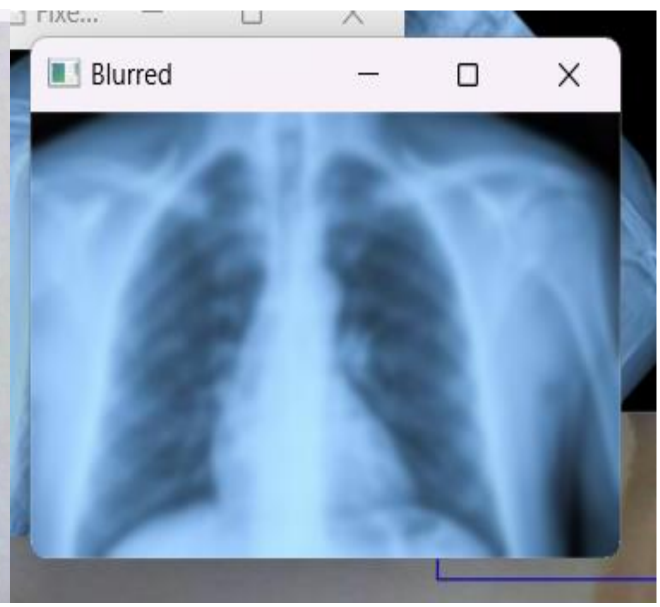







OpenCV Rotation



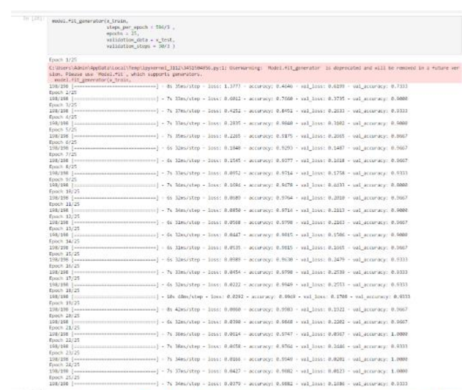

Blurred

9.RESULT:

9.1 PERFORMANCE METRICS:

S.No.	Parameter	Values	Screenshot

1.	Model Summary	`conv2d (Conv2D) - 320 max_pooling2d (MaxPooling2D) - 0 conv2d_1 (Conv2D) - 9248 max_pooling2d_1 (MaxPooling2D) - 0 flatten (Flatten) - 0 dense (Dense) - 802944 dense_1 (Dense) - 774`
		
		`================================`
		`Total params: 813,286`
		`Trainable params: 813,286`
		`Non-trainable params: 0`
			

2.	Accuracy	Training Accuracy - 98.82% Validation Accuracy – 93.33%	
3.	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	NA

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.

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

- source code
- demo link: https://youtu.be/WVOowdy5TEc
- https://github.com/IBM-EPBL/IBM-Project-13993-1659538539