PROJECT REPORT

A GESTURE-BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

submitted by

Team ID : PNT2022TMID37590

Srinath.S - (312019205036)

Frahison.T - (312019205003)

Mohan Raj.A - (312019205016)

Srinivasan.M - (312019205037)

TABLE OF CONTENT

1. INTRODUCTION

1.1 Project Overview

In this project we use gestures to browse images obtained during radiology. Gesture refers to nonverbal of communication made using hands. A major challenge involved in this process is to provide doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work. Keyboards and pointing device, such as a mouse, are today's principle 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. In this paper, we suggest the use of hand gestures as an alternative to existing interface techniques, offering the major advantage of sterility. Humans can recognize body and sign language easily. This is possibly due to the combination of vision and synaptic interactions that were formed 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 A Gesture-based Tool for Sterile Browsing of Radiology Images, First the model is trained pre trained on the images of different hand gestures, such as a showing number 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 are blurred; 2, image is resized; 3, image is rotated etc.

1.2 Purpose

- It can be used to interact with the application from a distance without any physical interaction with the keyboard or mouse.
- By using finger moments over a short period, the gesture tool can recognize the natural way of communicating between the human and computer.
- This gesture-based project helps a lot of doctors to perform their tasks more effectively.
- As the doctors need not to move anywhere during the entire operation, since all the commands were performed using hand gestures.

2. LITERATURE SURVEY

This paper's primary goal was to enhance the sterile browsing of radiology images. To avoid difficulties a gesture interface is developed for users, such as doctors/surgeons, to browse medical images in a sterile medical environment. A vision-based gesture capture system interprets user’s gestures in real-time to manipulate objects in an image visualization environment. The gesture system relies on real-time robust tracking of the user’s hand based in a motion fusion model.

Dynamic navigation gestures are translated to commands based on their relative positions on the screen. A state machine switches between other gestures such as zoom, blurred and rotate, as well as a sleep state. Performance evaluation included gesture recognition accuracy, task learning, and rotation accuracy. Fast task learning rates were found with convergence after ten trials. A beta test of a system prototype was conducted during a live brain biopsy operation, where neurosurgeons were able to browse through MRI images of the patient’s brain using the sterile hand gesture interface. The surgeons indicated the system was easy to use and fast with high overall satisfaction.

For any system the first step is to collect the necessary data to accomplish a specific task. For hand posture and gesture recognition system different technologies are used for acquiring input data. By tracking the motion or the movement of hand this project can fulfil the criteria of the user's need. The operation of the gesture interface was tested at the Washington Hospital Centre in Washington, DC. Two operations were observed in the hospital’s neurosurgery department and insights regarding the suitability of a hand gesture system was obtained. To our knowledge, this is the first time that a hand gesture recognition system was successfully implemented in an “in vivo” neurosurgical biopsy. 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.

We are now considering the addition of a body posture recognition system to increase the functionality of the system, as well as visual tracking of both hands to provide a richer set of gesture commands. This system serves as an aid for the patients and the doctors in carrying out certain primary functions without any physical contact which is the main reason for the transmission of any kind of microbes. The system mentioned in the paper would bring about a huge change in maintaining hygiene and safety in the premises of the hospital.

2.1 Existing Problem

- While accessing the camera some issues may happen like the system can’t be able to perform the faster.
- In order to replicate the skills in computer, the user must be able to separate objects of interest in image, capture technology and classification technique are more appropriate among others.

2.2 References

1. Schultz M, Gill J, Zubairi S, Huber R, Gordin F. “Bacterial contamination of computer keyboards in a teaching hospital,” *Infect Control Hosp. Epidemiol* 2003; 4(24):302-303.

2. Nishikawa A, Hosoi T, Koara K, Negoro D, Hikita A, Asano S, Kakutani H, Miyazaki F, Sekimoto M, Yasui M, Miyake Y, Takiguchi S, Monden M. “Face MOUSe: A Novel Human-Machine Interface for Controlling the Position of a Laparoscope,” *IEEE Trans. on Robotics and Automation* 2003; 19(5):825-841.

3. Smith KR, Frank KJ, Bucholz RD. “The NeuroStation- a highly accurate, minimally invasive solution to frameless stereotatic neurosurgery,” *Comput Med Imaging Graph* 1994; 18:247-256.

4. Graetzel C, Fong TW, Grange S, Baur C. “A non-contact mouse for surgeon-computer interaction,” *Technol Health Care* 2004;12(3):245-257.

5. Kuno Y, Murashima T, Shimada N, Shirai Y. “Intelligent Wheelchair Remotely Controlled by Interactive Gestures.” *Proceedings of 15th International Conference on Pattern Recognition* 2000; 4:672-675.

6. Starner T, Auxier J, Ashbrook D, Gandy M. “The Gesture Pendant: A Self-illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring” *Fourth Intl. Symp Wearable Comp* 2000:87-94.

7. Wachs JP, Stern HI, Edan Y, et al. “Real-Time Hand Gesture Interface for Browsing Medical Images” *Int. J Intel. Comp. Med. Sci. Image Proc* 2007;1(3):175-185.

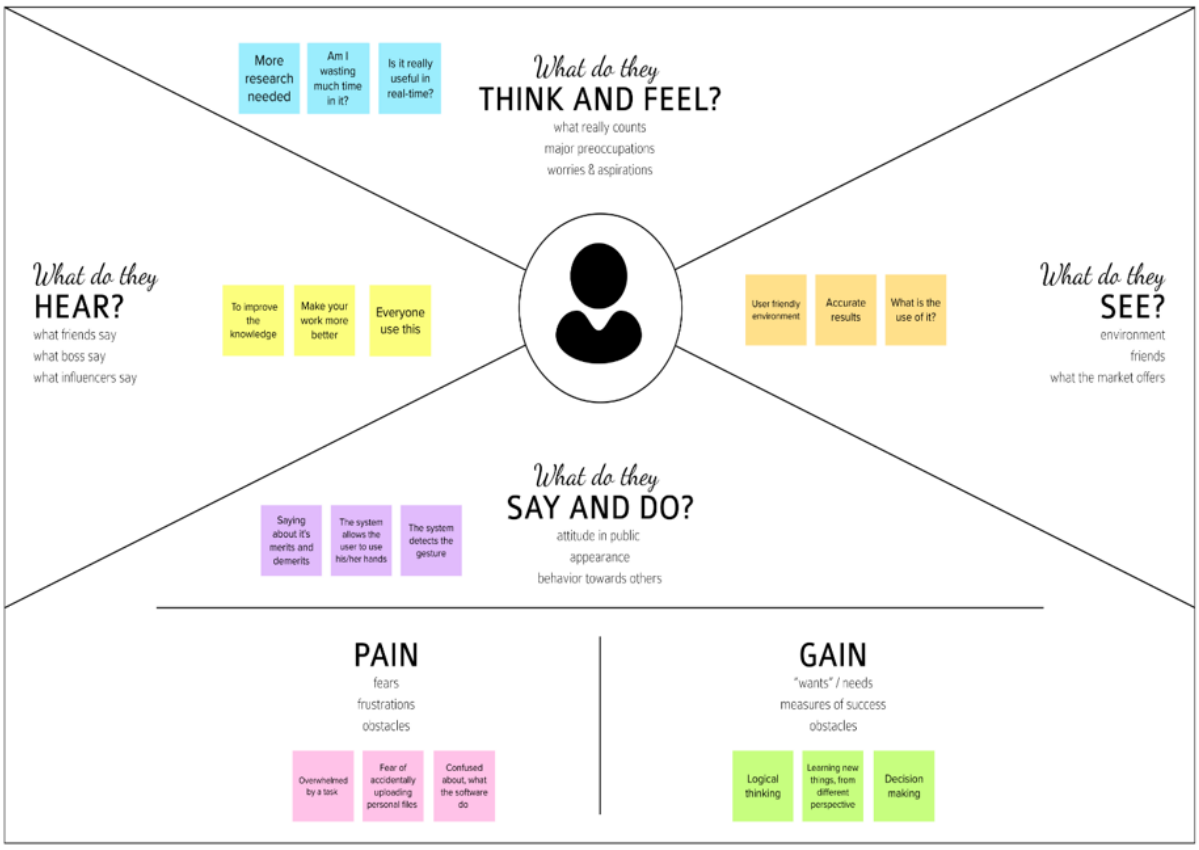
8. Lewis JR. Psychometric evaluation of an after scenario questionnaire for computer usability studies: The ASQ *SIGCHI Bulletin* 1991;23:78-81.

2.3 Problem Statement Definition

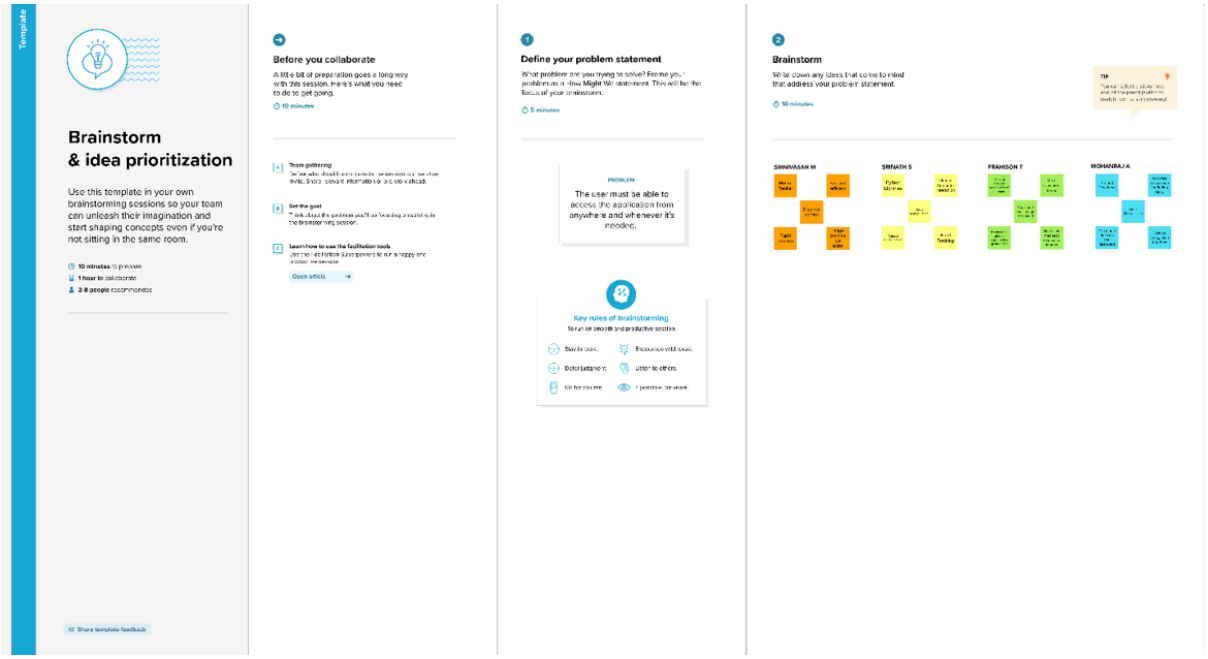
- The webcams must be able to recognize the motion gestures which are already trained to the machine.
- The user must be able to access the application from anywhere and whenever it's needed.
- The user must be able to upload images of various sizes and able to blur, resize and the images must be rotated respectively. This tool must be able to interact with humans and able to understand the symbols.
- The machine must be capable of performing several operations at the same time without any interruptions.

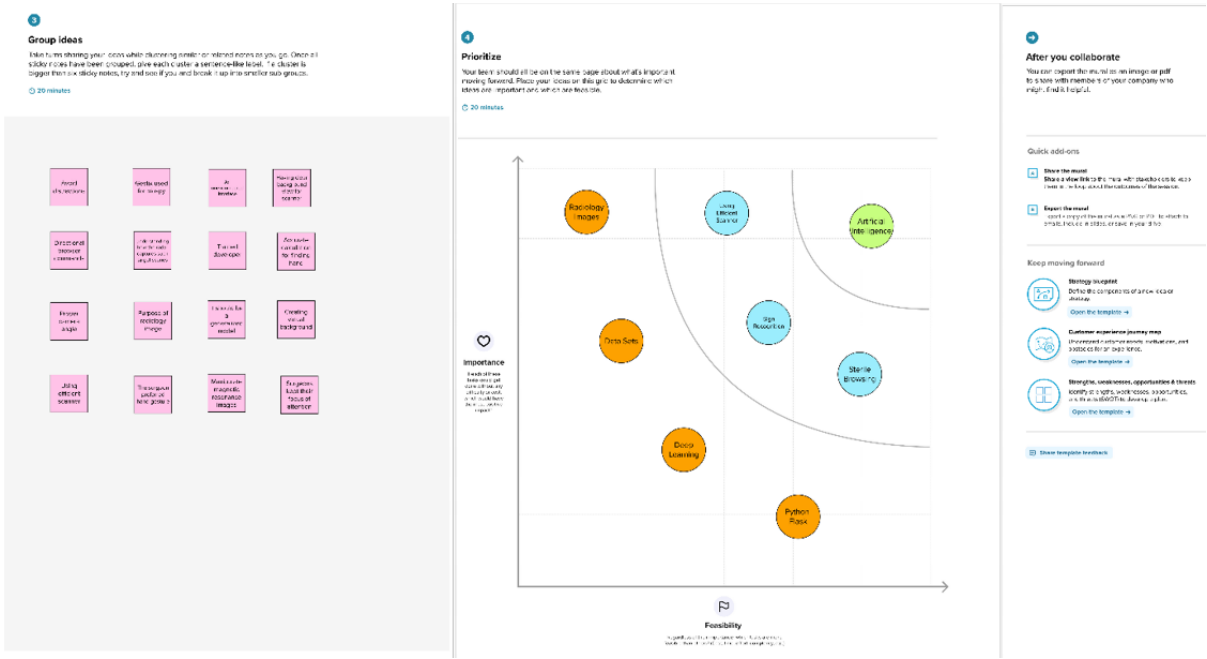
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

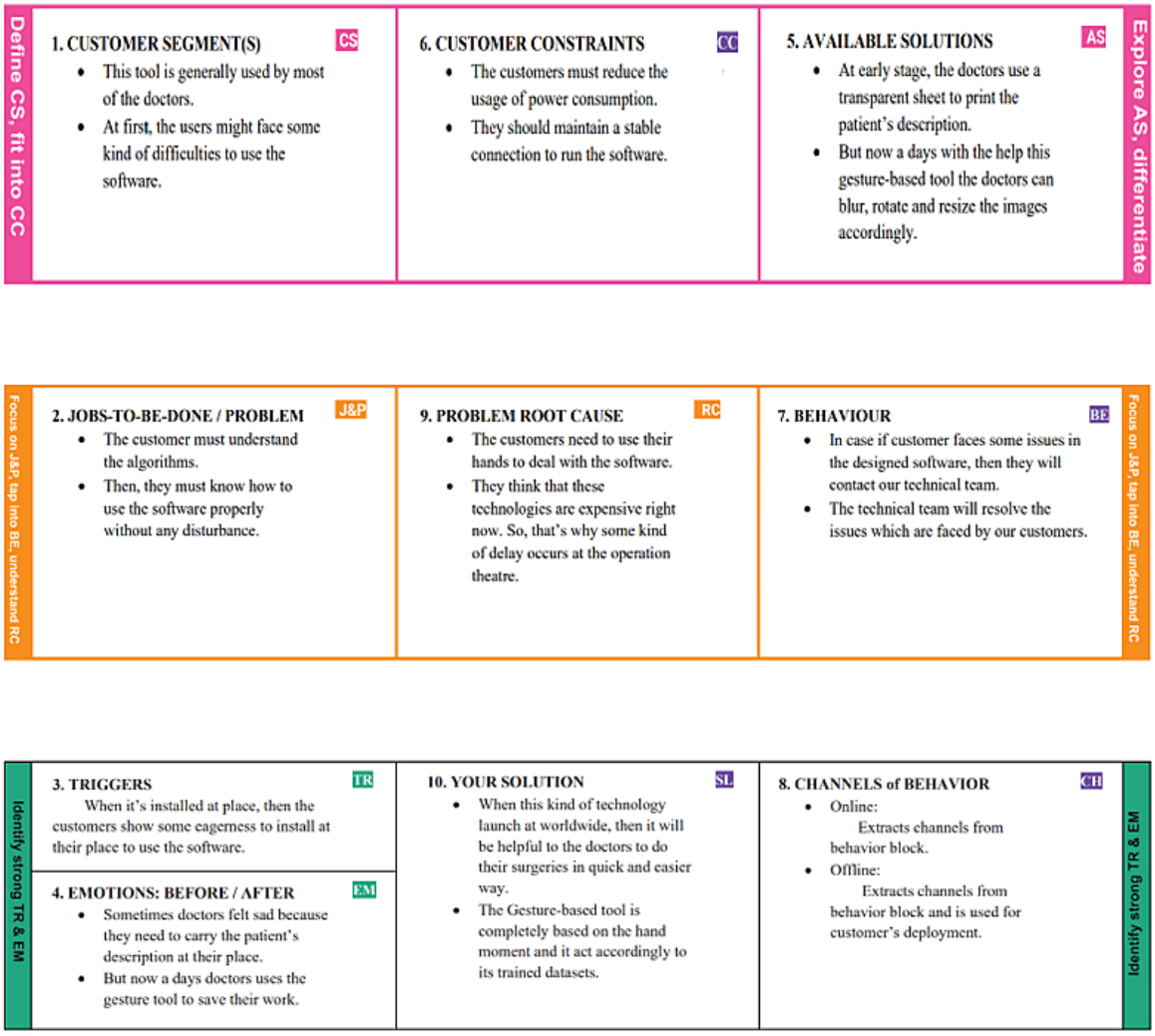




3.3 Proposed Solution

SL. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A doctor needs a way to browse the radiological images during surgery so, that they don't need to reach report/Lab room every time.
2.	Idea / Solution description	Doctor can make use of hand gesture to move or control the images.
3.	Novelty / Uniqueness	In this method unlike other methods of non-verbal communication, gesture do not cause loss of concentration in operation theatre. It performs better in detecting pattern in images.
4.	Social Impact / Customer Satisfaction	It co-operates social responsibility by providing better solution to patient's health and it also helps professionals to browse images without having direct contact with system which avoids the harmful rays.
5.	Business Model (Revenue Model)	Cost efficient to deploy this Software for health care department as well as in hospitals and can collaborate with government for health awareness camps.
6.	Scalability of the Solution	Better execution in accurate results, sensitivity, system architecture design and flexibility of the software.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration is done through the web application page to login.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Authentication	It is through password authentication protocal.
FR-4	Hand detection	Filtering of hand from image capturing device.
FR-5	Model rendering	When the user uploads the gesture, the algorithm should start processing its task.
FR-6	Reporting	If any issues are faced by the customer or user, it will be directly notified to the developer.

4.2 Non-Functional requirements

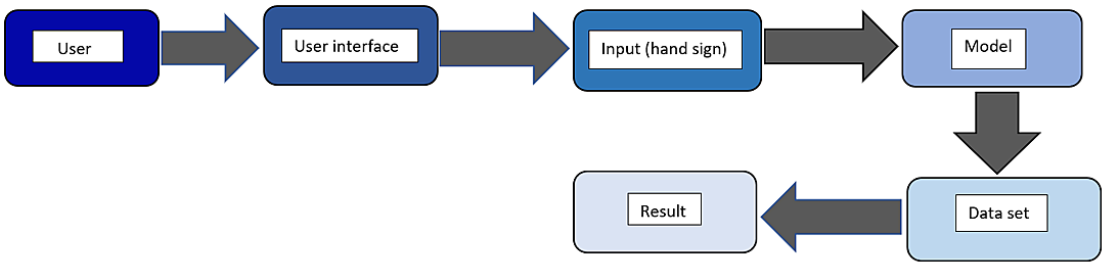
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	It can be used for all users easily and also understandable for non- technical users to communicate with computer via hand sign.
NFR-2	Security	Accessible only in secure networks with administrative permissions, so there is less chance of security breach.
NFR-3	Reliability	Its operatable under all condition and also, we can communicate with computer.
NFR-4	Performance	The performance of the software is high because the speed and accuracy are high. It also upgrades the lifestyle of human beings controlling things via hand signs.
NFR-5	Availability	When the gesture is available then only the application works. This application is only available in surgery rooms.
NFR-6	Scalability	In future we can develop the vehicles that would being controlled by hand gestures.

5. PROJECT DESIGN

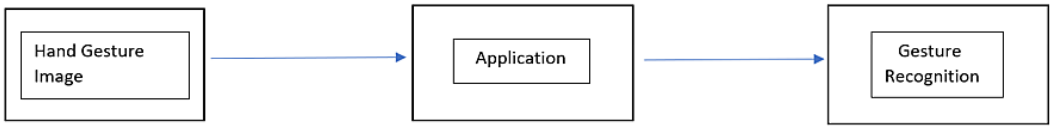
5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

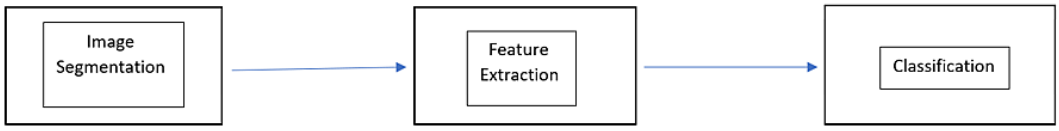
Simplified:



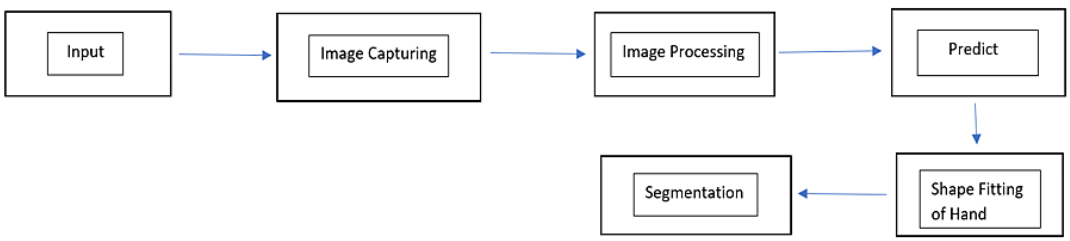
Data Flow Diagram-Level 0



Data Flow Diagram-Level 1



Data Flow Diagram-Level 2



5.2 Solution & Technical Architecture

Solution Architecture:

- ➤ User interacts with the UI (User Interface) to upload the image as input.
- ➤ Depending on the different gesture inputs different operations are applied to the input image.
- ➤ The image can be resized, blur, flip and rectangle.
- ➤ Once model analyses the gesture, the prediction with operation applied on image is showcased on the UI.
- ➤ Better execution in accurate results, sensitivity, system architecture design and flexibility of the software.

Technology Architecture Diagram:

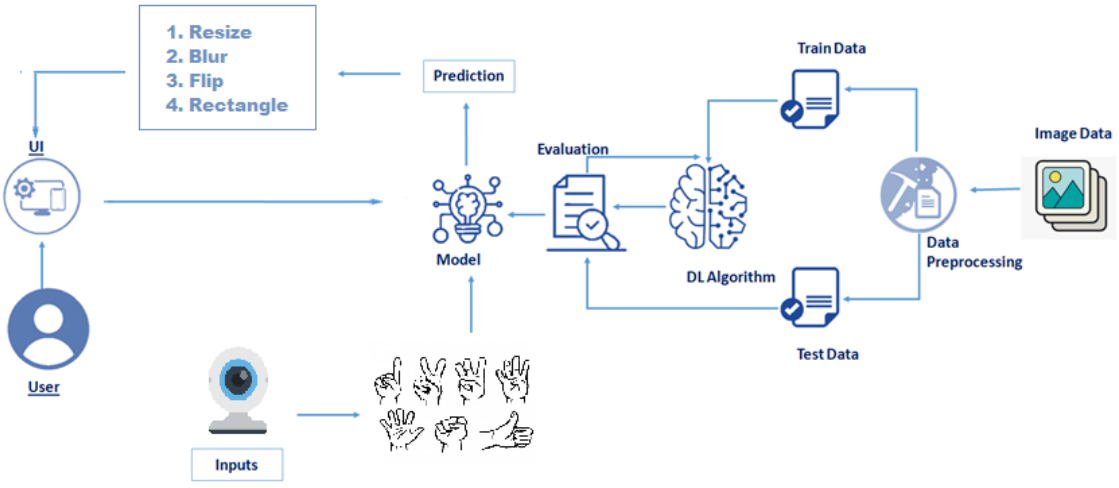


Table-1: Components & Technologies:

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	To develop the project variety of frameworks, libraries and supports are required.	Java / Python
3.	Application Logic-2	Helps to convert the hand gestures and communicates with the computer.	IBM Watson STT service
4.	Application Logic-3	It provides accurate answers after recognizing the human hand gesture.	IBM Watson Assistant
5.	Database	It can be numerical, time series data.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud.	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage will be reliable, scalable, fast and flexible.	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Access information in the cloud.	IBM Weather API, etc.
9.	External API-2	Access the information for data driven decision making.	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model deals with various algorithm for the implementation.	Image Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud ➢ Local Server Configuration ➢ Cloud Server Configuration	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The open-source frameworks used in project are	Py Torch, Tensor flow, Flask.
2.	Security Implementations	The security / access controls implemented, use of firewalls etc.	Other security related software's.
3.	Scalable Architecture	The scalability of architecture (3 – tier, Micro-services)	Data models, operate at size, consistency and speed.
4.	Availability	The availability of application (e.g. use of load balancers, distributed servers etc.)	Image recognition and real time captioning.
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Full and effective participation, equality of opportunity, accessibility.

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer[surgeon] (user)	Launch	USN-1	As a user, I can launch the webpage to upload and manipulate the scan images	I can access the webpage	High	Sprint-4
		USN-2	As a user, I can use different web browser	I can access the webpage using different web browser	High	Sprint-1
Administrator		USN-1	Access the database	Database Management	High	Sprint-3
		USN-2	Server crash, database recovery	Resolve the errors or issue, recover the last data from the database	High	Sprint-5
Customer care executive	Availability	USN-1	Interpret and recognize gesture inaccurately	Webcam detection	Medium	Sprint-5
		USN-2	When the website is unresponsive or an internal error occurs in the Website	Webpage is unresponsive	Medium	Sprint-5
	Predict	USN-3	As a user I can turn on the camera using predict button	I can turn on camera for prediction	High	Sprint-3
		USN-4	Predicating the images using Hand Gesture	I can resize, blur, and flip my image using my hand gesture	High	Sprint-3
		USN-5	I can give a gesture of raised fist and it recognize	I can get my fixed resized image	High	Sprint-4
		USN-6	I can show my index finger	I can get a rectangular image	High	Sprint-4
		USN-7	I can show my index finger middle finger and ring finger at once	I can get my image blurred	High	Sprint-4

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

User Story Number	User Story / Task	Story Points	Priority	Team Members
USN-1	To analyse the hand gesture and to set the integrated camera to collect the image and observe the injured patient.	2	High	Frahison T Mohanraj A Srinath S Srinivasan M
USN-2	The collected data are categorized on the basis of parameters set to identify. The model building libraries and initializing the model, Adding CNN layers and dense layers to configure the learning processes by storing the datasets in server.	1	High	Frahison T Mohanraj A Srinath S Srinivasan M
USN-3	The main task is to check that the model is efficient to work in real time. Therefore, smallest of error decoded needed to be corrected to avoid future lags	2	Medium	Frahison T Mohanraj A Srinath S Srinivasan M
USN-4	The model after testing all its functionalities is been implemented at Hospital in the surgery room to get quick responses from the model	2	High	Frahison T Mohanraj A Srinath S Srinivasan 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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	15	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	10	14 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	5	20 Nov 2022

7.CODING & SOLUTIONING

7.1 Feature

```
result = model.predict(test_image.reshape(1, 64, 64, 1))
prediction = {'ZERO': result[0][0],
              'ONE': result[0][1],
              'TWO': result[0][2],
              'THREE': result[0][3],
              'FOUR': result[0][4],
              'FIVE': result[0][5]}
# Sorting based on top prediction
prediction = sorted(prediction.items(), key=operator.itemgetter(1), reverse=True)

# Displaying the predictions
cv2.putText(frame, prediction[0][0], (10, 120), cv2.FONT_HERSHEY_PLAIN, 1, (0,255,255), 1)
cv2.imshow("Frame", frame)

#loading an image
image1=cv2.imread(file_path)
if prediction[0][0]=='ONE':
    resized = cv2.resize(image1, (200, 200))
    cv2.imshow("Fixed Resizing", resized)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("1"):
        cv2.destroyWindow("Fixed Resizing")

elif prediction[0][0]=='ZERO':
    cv2.rectangle(image1, (480, 170), (650, 420), (0, 0, 255), 2)
    cv2.imshow("Rectangle", image1)
    cv2.waitKey(0)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("0"):
        cv2.destroyWindow("Rectangle")
```

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

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

    elif prediction[0][0]=='FOUR':
        resized = cv2.resize(image1, (400, 400))
        cv2.imshow("Fixed Resizing", resized)
        key=cv2.waitKey(3000)
        if (key & 0xFF) == ord("4"):
            cv2.destroyWindow("Fixed Resizing")

    elif prediction[0][0]=='FIVE':
        gray = cv2.cvtColor(image1, cv2.COLOR_RGB2GRAY)
        cv2.imshow("OpenCV Gray Scale", gray)
        key=cv2.waitKey(3000)
        if (key & 0xFF) == ord("5"):
            cv2.destroyWindow("OpenCV Gray Scale")
```

8.TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	BUG ID	Executed By
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	1) Open the page 2) Check if all the UI elements are displayed	1700.1: 8000	The Home page must be displayed properly	Working as expected	PASS		Frahiem T Mohamed A
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	1) Open the page in a specific device 2) Check if all the UI elements are displayed properly 3) Repeat the above steps with different device sizes	– Screen Sizes –2.560x 1801 1440 x 970 1024 x 940 768 x 630 320 x 630	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1800 and 786 x 630	FAIL	BUG_HP_001	Frahiem T Mohamed A
HP_TC_003	Functional	Home Page	Check if the page redirects to the result page once the input is given	1) Open the page 2) Click on select button 3) Click on web camera 4) Check if the page redirects	Camera feed	The page should redirect to the result page	Working as expected	PASS		Srinath S Srinivasan M
BE_TC_001	Functional	Backend	Check if all the routes are working properly	1) Go to Home Page 2) Click on web camera 3) Check the results page	Camera feed	All the routes should properly work	Working as expected	PASS		Srinath S Srinivasan M
M_TC_001	Functional	Model	Check if the model can handle various image	1) Open the page in a specific device 2) Click on Web Camera 3) Repeat the above steps with different images	Camera feed	The model should rescale the image and predict the results	Working as expected	PASS		Srinath S Srinivasan M
M_TC_002	Functional	Model	Check if the model predicts the disaster	1) Open the page 2) Click on Web Camera 3) Check the results	Camera feed	The model should predict the disaster	Working as expected	PASS		Srinath S Srinivasan M
M_TC_003	Functional	Model	Check if the model can handle complex input	1) Open the page 2) Click on Web Camera 3) Check the results	Complex camera feed	The model should predict the disaster in the complex feed	The model fails to identify it since the model is not built to handle such data	FAIL	BUG_M_001	Srinath S Srinivasan M
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	1) Open the page 2) Click on Web Camera 3) Check if all the UI elements are displayed properly	Camera feed	The Result page must be displayed properly	Working as expected	PASS		Frahiem T Mohamed A
RP_TC_002	UI	Result Page	Check if the result is displayed properly	1) Open the page 2) Click on Web Camera 3) Check if the result is displayed	Camera feed	The result should be displayed properly	Working as expected	PASS		Frahiem T Mohamed A
RP_TC_003	UI	Result Page	Check if the other predictions are displayed properly	1) Open the page 2) Click on Web Camera 3) Check if all the other predictions are displayed	Camera feed	The other predictions should be displayed properly	Working as expected	PASS		Frahiem T Mohamed A

8.2 User Acceptance Testing

PURPOSE OF THE DOCUMENT

The purpose of this document is to briefly explain the test coverage and open issues of the Handwritten Digit Recognition project at the time of the release to User Acceptance Testing (UAT).

DEFECT ANALYSIS

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

TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

9.RESULTS

9.1 Performance Metrics

Locust Test Report

During: 11/14/2022, 10:54:06 AM - 11/14/2022, 10:56:49 AM

Target Host: http://127.0.0.1:5000

Script: locust.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	151	0	7	3	18	6975	0.9	0.0
GET	/image1	156	0	6	3	25	7090	1.0	0.0
GET	/intro	159	0	6	3	18	8317	1.0	0.0
GET	/predict	42	4	15431	2982	95299	6335	0.3	0.0
	Aggregated	**508**	**4**	**1281**	**3**	**95299**	**7377**	**3.1**	**0.0**

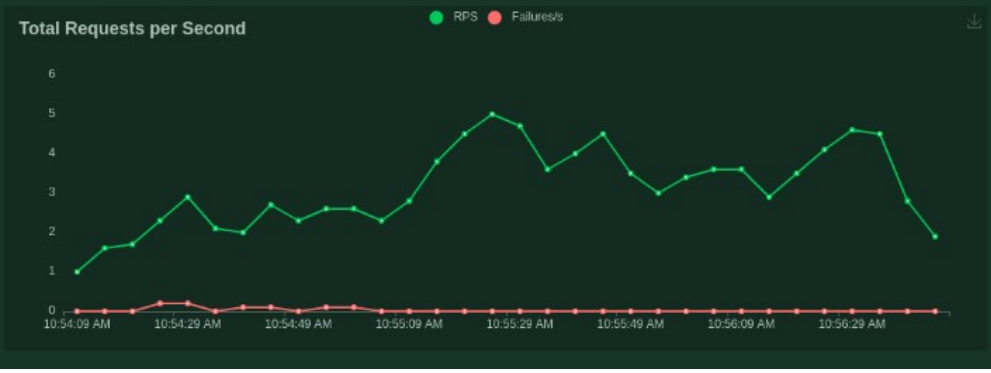
Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	6	7	9	10	11	12	14	18
GET	/image1	5	6	7	8	10	11	16	26
GET	/intro	6	6	7	9	11	11	18	18
GET	/predict	7500	11000	17000	21000	40000	49000	95000	95000
	Aggregated	**6**	**7**	**9**	**10**	**13**	**6300**	**31000**	**95000**

Failures Statistics

Method	Name	Error	Occurrences
GET	/predict	500 Server Error: INTERNAL SERVER ERROR for url: http://127.0.0.1:5000/predict	4

Charts



Final ratio

Ratio per User class

- 100.0% AppUser
  - 25.0% home
  - 25.0% intro
  - 25.0% image1
  - 25.0% predict

Total ratio

- 100.0% AppUser
  - 25.0% home
  - 25.0% intro
  - 25.0% image1
  - 25.0% predict

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.
- Such systems are difficilt to develop because of the complexity and the cost of implementation.
- As each gesture is assigned a specific control command , this system is not platform independent since certain control commands vary as the operating system varies.

11.CONCLUSION

In this project we developed a tool which recognises hand gestures and enables doctors to browse through radiology images using these gestures. This enables doctors and surgeons to maintain the sterility as they would not have to touch any mouse or keyboard to go through the images. This tool is also easy to use and is quicker than the regular method of using mouse/keyboard. It can be used regardless of the users location since they don't have to be in contact with any device. It also does not require the user to have any device on them to use it. Further this technology can be extended to other industries like it can be used by presenters, by teachers for show images in the classroom, etc.

12.FUTURE SCOPE

- The tool can be made quicker by increasing the recognition speed.

- More number of gestures can be added thereby increasing this tool's functionality and useability for different purposes.

- Tracking of both hands can be added to increase the set of commands.

- Voice commands can also be added to further increase the functionality.

13.APPENDIX

13.1 Source Code

MODEL CREATION

Model Training

Importing packages

In [1]:
```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten, Dropout
from tensorflow.keras.layers import Conv2D,MaxPooling2D
from keras.preprocessing.image import ImageDataGenerator
```

Image Data Argumentation

In [2]:
```
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
```

Loading Our Data And Perform Data Argumentation

In [3]:
```
x_train = train_datagen.flow_from_directory(r'D:\IBM Project\Dataset\train', target_size=(64, 64), batch_size=3, color_mode='grayscale',
x_test = test_datagen.flow_from_directory(r'D:\IBM Project\Dataset\test', target_size=(64, 64), batch_size=3, color_mode='grayscale', cla
```

```
Found 594 images belonging to 6 classes.
Found 30 images belonging to 6 classes.
```

In [4]:
```
print(x_train.class_indices)
```

```
{'0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5}
```

Initializing The Model

In [5]:
```
model=Sequential()
```

Adding CNN Layers

In [6]:
```
model.add(Conv2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

In [7]:
```
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

In [8]:
```
model.add(Flatten())
```

Adding Dense Layers

In [9]:
```
model.add(Dense(units=512, activation='relu'))
```

In [10]:
```
model.add(Dense(units=6, activation='softmax'))
```

In [11]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        320

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

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

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

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 512)               3211776

 dense_1 (Dense)             (None, 6)                 3078

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

Configure The Learning Process

In [12]:
```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Train The Model

In [13]:
```
model.fit_generator(x_train,
                    steps_per_epoch =594/3,
                    epochs=25,
                    validation_data=x_test,
                    validation_steps=30/3)
```

```
C:\Users\srina\AppData\Local\Temp/ipykernel_9384/1173897450.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,
Epoch 1/25
198/198 [==============================] - 19s 88ms/step - loss: 1.3609 - accuracy: 0.4764 - val_loss: 0.7358 - val_accuracy: 0.6667
Epoch 2/25
198/198 [==============================] - 17s 83ms/step - loss: 0.5948 - accuracy: 0.7525 - val_loss: 0.5594 - val_accuracy: 0.7667
Epoch 3/25
198/198 [==============================] - 15s 74ms/step - loss: 0.3788 - accuracy: 0.8468 - val_loss: 0.3211 - val_accuracy: 0.8333
Epoch 4/25
198/198 [==============================] - 16s 79ms/step - loss: 0.2756 - accuracy: 0.8805 - val_loss: 0.4424 - val_accuracy: 0.8000
Epoch 5/25
198/198 [==============================] - 15s 73ms/step - loss: 0.2200 - accuracy: 0.9242 - val_loss: 0.1211 - val_accuracy: 0.9667
Epoch 6/25
198/198 [==============================] - 14s 72ms/step - loss: 0.1728 - accuracy: 0.9377 - val_loss: 0.1738 - val_accuracy: 0.9333
Epoch 7/25
198/198 [==============================] - 13s 67ms/step - loss: 0.1086 - accuracy: 0.9630 - val_loss: 0.3307 - val_accuracy: 0.9667
Epoch 8/25
198/198 [==============================] - 16s 79ms/step - loss: 0.1218 - accuracy: 0.9411 - val_loss: 0.2783 - val_accuracy: 0.9667
Epoch 9/25
198/198 [==============================] - 14s 68ms/step - loss: 0.1172 - accuracy: 0.9646 - val_loss: 0.1371 - val_accuracy: 0.9667
Epoch 10/25
198/198 [==============================] - 15s 76ms/step - loss: 0.0850 - accuracy: 0.9731 - val_loss: 0.2077 - val_accuracy: 0.9667
Epoch 11/25
198/198 [==============================] - 13s 67ms/step - loss: 0.0314 - accuracy: 0.9933 - val_loss: 0.2819 - val_accuracy: 0.9667
Epoch 12/25
198/198 [==============================] - 14s 69ms/step - loss: 0.0698 - accuracy: 0.9731 - val_loss: 0.3276 - val_accuracy: 0.9667
Epoch 13/25
198/198 [==============================] - 14s 68ms/step - loss: 0.0671 - accuracy: 0.9764 - val_loss: 0.3040 - val_accuracy: 0.9667
Epoch 14/25
```

Save The Model

In [14]:
```
model.save('gesture.h5')
```

In [15]:
```
model_json = model.to_json()
with open("model-bw.json", "w") as json_file:
    json_file.write(model_json)
```

Model Testing

In [16]:
```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model = load_model("gesture.h5")
```

In [17]:
```
img = image.load_img(r'D:\IBM Project\Dataset\test\1\1.jpg', grayscale=True, target_size=(64,64))
x = image.img_to_array(img)
x.shape
```

```
C:\Users\srina\anaconda3\lib\site-packages\keras\utils\image_utils.py:409: UserWarning: grayscale is deprecated. Please use color_mode = "grayscale"
  warnings.warn(
```

Out[17]:
```
(64, 64, 1)
```

In [18]:
```
x = np.expand_dims(x,axis=0)
x.shape
```

Out[18]:
```
(1, 64, 64, 1)
```

In [19]:
```
pred_x = model.predict(x)
pred_x=np.argmax(pred_x,axis=1)
pred_x
```

```
1/1 [==============================] - 0s 201ms/step
```

Out[19]:
```
array([1], dtype=int64)
```

In [20]:
```
index=['0', '1', '2', '3', '4', '5']
result=str(index[pred_x[0]])
result
```

Out[20]:
```
'1'
```

In [21]:
```
import numpy as np
p=[]
for i in range(0,6):
  for j in range(0,5):
    img = image.load_img(r"D:\\IBM Project\\Dataset\test\\"+str(i)+"\\"+str(j)+".jpg", grayscale=True, target_size=(64,64))
    x = image.img_to_array(img)
    x = np.expand_dims(x,axis=0)
    pred = np.argmax(model.predict(x),axis=-1)
    p.append(pred)
print(p)
```

```
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 63ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 63ms/step
```

```
1/1 [==============================] - 0s 47ms/step
[array([0], dtype=int64), array([0], dtype=int64), array([0], dtype=int64), array([0], dtype=int64), array([0], dtype=int64), array([1],
dtype=int64), array([1], dtype=int64), array([1], dtype=int64), array([1], dtype=int64), array([1], dtype=int64), array([2], dtype=int6
4), array([2], dtype=int64), array([1], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), array([3], dtype=int64), array
([3], dtype=int64), array([3], dtype=int64), array([3], dtype=int64), array([3], dtype=int64), array([4], dtype=int64), array([4], dtype=
int64), array([4], dtype=int64), array([4], dtype=int64), array([4], dtype=int64), array([5], dtype=int64), array([5], dtype=int64), arra
y([5], dtype=int64), array([5], dtype=int64), array([5], dtype=int64)]
```

In [22]:

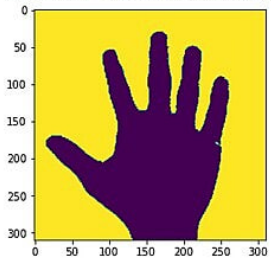
```
result = []
index=['0','1','2','3','4','5']
for i in p:
  result.append(index[i[0]])
print(result)
```

```
['0', '0', '0', '0', '0', '1', '1', '1', '1', '1', '2', '2', '1', '2', '2', '3', '3', '3', '3', '3', '4', '4', '4', '4', '4', '5', '5', '
5', '5', '5']
```

In [23]:

```
%pylab inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
imgs = mpimg.imread(r"D:\IBM Project\Dataset\test\5\0.jpg")
imgplot = plt.imshow(imgs)
plt.show()
```

```
Populating the interactive namespace from numpy and matplotlib
```



FLASK APP.PY

```
from flask import Flask,render_template,request
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
import operator
import cv2 # opencv library
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
from tensorflow.keras.models import load_model#to load our trained model
import os
from werkzeug.utils import secure_filename
```

```
app = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model(r'D:\IBM Project\front-end\FLASK\gesture.h5')
print("Loaded model from disk")


@app.route('/')# route to display the home page
def home():
    return render_template('home.html')#rendering the home page


@app.route('/intro') # routes to the intro page
def intro():
    return render_template('intro.html')#rendering the intro page

@app.route('/image1',methods=['GET','POST'])# routes to the index html
def image1():
    return render_template("launch.html")
```

```
@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def launch():
    if request.method == 'POST':
        print("inside image")
        f = request.files['image']
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        print(file_path)
        cap = cv2.VideoCapture(0)
        while True:
            _, frame = cap.read() #capturing the video frame values
            # Simulating mirror image
            frame = cv2.flip(frame, 1)
            # Got this from collect-data.py
            # Coordinates of the ROI
            x1 = int(0.5*frame.shape[1])
            y1 = 10
            x2 = frame.shape[1]-10
            y2 = int(0.5*frame.shape[1])
            # Drawing the ROI
            # The increment/decrement by 1 is to compensate for the bounding box
            cv2.rectangle(frame, (x1-1, y1-1), (x2+1, y2+1), (255,0,0) ,1)
            # Extracting the ROI
            roi = frame[y1:y2, x1:x2]

            # Resizing the ROI so it can be fed to the model for prediction
            roi = cv2.resize(roi, (64, 64))
            roi = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
            _, test_image = cv2.threshold(roi, 120, 255, cv2.THRESH_BINARY)
            cv2.imshow("test", test_image)
            # Batch of 1
```

```
result = model.predict(test_image.reshape(1, 64, 64, 1))
prediction = {'ZERO': result[0][0],
              'ONE': result[0][1],
              'TWO': result[0][2],
              'THREE': result[0][3],
              'FOUR': result[0][4],
              'FIVE': result[0][5]}
# Sorting based on top prediction
prediction = sorted(prediction.items(), key=operator.itemgetter(1), reverse=True)

# Displaying the predictions
cv2.putText(frame, prediction[0][0], (10, 120), cv2.FONT_HERSHEY_PLAIN, 1, (0,255,255), 1)
cv2.imshow("Frame", frame)

#loading an image
image1=cv2.imread(file_path)
if prediction[0][0]=='ONE':
    resized = cv2.resize(image1, (200, 200))
    cv2.imshow("Fixed Resizing", resized)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("1"):
        cv2.destroyWindow("Fixed Resizing")

elif prediction[0][0]=='ZERO':
    cv2.rectangle(image1, (480, 170), (650, 420), (0, 0, 255), 2)
    cv2.imshow("Rectangle", image1)
    cv2.waitKey(0)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("0"):
        cv2.destroyWindow("Rectangle")
```

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

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

    elif prediction[0][0]=='FOUR':
        resized = cv2.resize(image1, (400, 400))
        cv2.imshow("Fixed Resizing", resized)
        key=cv2.waitKey(3000)
        if (key & 0xFF) == ord("4"):
            cv2.destroyWindow("Fixed Resizing")

    elif prediction[0][0]=='FIVE':
        gray = cv2.cvtColor(image1, cv2.COLOR_RGB2GRAY)
        cv2.imshow("OpenCV Gray Scale", gray)
        key=cv2.waitKey(3000)
        if (key & 0xFF) == ord("5"):
            cv2.destroyWindow("OpenCV Gray Scale")
```

```
            else:
                continue


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


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

if __name__ == "__main__":
    # running the app
    app.run(debug=False)
```

HOME.HTML

```
<html>
<body>
<h1 style="color: rgb(193, 207, 207);">
  <table style="width:100%">
    <tr>
      <th><img src="https://media.tenor.com/images/30169e4a670daf12443df7d2dd140176/tenor.gif" alt="NOT AVAILABLE" width="300" height="300"></th>
    </tr>
  </table>
    <br>
    <span>HAND   GESTURE   RECOGNITION</span>
  </br>
    <span>OF</span>
  </br>
    <span>RADIOLOGY   IMAGES</span>
  </br>
    <span>THROUGH</span>
  </br>
    <span>STERILE   BROWSING</span>
    </h1>
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color: #c4dfd7; padding-top:1%;padding-left:5%;"><b>HAND GESTURE RECOGNITION</b></div>
  <div class="topnav-right"style="padding-top:0.5%;color:white">
    <a  class="active" href="{{ url_for('home')}}"><u>Home</u></a>
    <a  class="active" href="{{ url_for('intro')}}">Introduction</a>
    <a  class="active" href="{{ url_for('image1')}}">Launch</a>
  </div>
</div>
</body>
</html>
```

INTRO.HTML

```
<html>
<body>
  <h1>INTRODUCTION</h1>
<h1>
  <span> Hand Gesture </span> <span> recognition system </span> <span> provides us </span > <span> an innovative,</span> <span>natural,</span> <span>  user friend
  <span> which is more </span> <span> familiar to the </span> <span>human beings. </span > <span> In our project, </span> <span> the hand region </span> <span> is
<span> Region of intrest. </span> <span> Then, </span> <span> we will be </span > <span>predicting the labels </span> <span> based on the </span> <span> CNN train
<span> we apply if conditions </span> <span> to control some of the actions </span> <span>like </span > <span>reshaping , blur, flip of the given image.</span>

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

<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color: #c1e2d9; padding-top:1%;padding-left:5%;">Hand Gesture System</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a class="active" href="{{ url_for('intro')}}"><u>Introduction</u></a>
    <a class="active" href="{{ url_for('image1')}}">Launch</a>
  </div>
</div>
</body>

</html>
```

LAUNCH.HTML

```
<html>
<body>
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color: #c1e2d9; padding-top:1%;padding-left:5%;">Hand Gesture System</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a class="active" href="{{ url_for('intro')}}">Introduction</a>
    <a class="active" href="{{ url_for('image1')}}"><u>Launch</u></a>
  </div>
</div>
<br>

  <div1 ><h1><font color="Black" size="6" font-family="Roboto">Hand Gesture Recognition</h1><br>
  <p><i><font color="Black" size="4" fonr-family="sans-serif"></i>Provide an image for which you want to perform various operations</p>
  <br>
        <div>
          <h4>Upload Image Here</h4>
      <form action = "http://localhost:5000/" id="upload-file" method="POST" enctype="multipart/form-data">
        <label for="imageUpload" class="upload-label">
          Choose...
        </label>
        <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg,.pdf">
      </form>
            <center>
      <div class="image-section" style="display:none;">
        <div class="img-preview">
          <div id="imagePreview">
          </div>
        </div>
        <div>
```

```
            </div>
            <div>
              <button type="button" class="btn btn-info btn-lg " id="btn-predict">Predict!</button>
            </div>
          </div>
          <div class="loader" style="display:none;"></div>
              </center>
      </div>
      </div1>

          <footer>
      <script src="{{ url_for('static', filename='main.js') }}" type="text/javascript"></script>
  </footer>

  </html>
```

LAUNCH.CSS

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

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: cover;
    background-repeat: no-repeat;
    background-position: center;
}

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

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

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

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

LAUNCH.JS

```javascript
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();

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

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

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

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

        // Make prediction by calling api /predict
        $.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').html(data);
                console.log('Success!');
            },
        });
    });

});
```

13.2 GitHub & Project Demo Link

GitHub LInk

https://github.com/IBM-EPBL/IBM-Project-1519-1658393990

Project Demo Video Link

https://drive.google.com/file/d/188Zjjt0jDLrpy6k6UeIO6IN4efKy9f6j/view?usp=share_link