A GESTURE – BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

NALAIYA THIRAN PROJECT BASED LEARNING

on

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

A PROJECT REPORT

PRAISON SOLOMON V (19104129)

PUGAZHENDHI N (19104132)

VENKADESH R (19104169)

VIGNESH K (19104170)

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

HINDUSTHAN COLLEGE OF ENGINEERING AND TECHOLOGY

Approved by AICTE, New Delhi, Accredited with 'A' Grade by NAAC

(An Autonomous Institution, Affiliated to Anna University, Chennai)

COIMBATORE – 641 032

HINDUSTHAN COLLEGE OF ENGINEERING AND TECHOLOGY

Approved by AICTE, New Delhi, Accredited with 'A' Grade by NAAC

(An Autonomous Institution, Affiliated to Anna University, Chennai)

COIMBATORE – 641 032

INTERNAL MENTOR

TKP Rajagopal,

Assistant

Professor,

Department of Computer Science and

Engineering, Hindusthan College of

Engineering and Technology, Coimbatore-

641 032

INDUSTRY MENTOR

Pradeepthi

IBM

November 2022

TABLE OF CONTENTS

Abstract

The 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 times. 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 electronic medical record (EMR) database. Navigation and other gestures are translated to commands based on their temporal trajectories, through video capture. “Gestix” was tested during a brain biopsy procedure. 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 from two usability tests provide insights and implications regarding human-computer interaction based on nonverbal conversational modalities.

CHAPTER 1

INTRODUCTION

Computer information technology is increasingly penetrating into the hospital domain. A major challenge involved in this process is to provide doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work. Keyboards and pointing devices, such as a mouse, are today's principal method of human—computer interaction. However, the use of computer keyboards and mice by doctors and nurses in intensive care units (ICUs) is a common method for spreading infections. In this paper, we suggest the use of hand gestures as an alternative to existing interface techniques, offering the major advantage of sterility. Even though voice control also provides sterility, the noise level in the operating room (OR) deems it problematic.

In this work we refer to gestures as a basic form of non-verbal communication made with the hands. Psychological studies showed that young children use gestures to communicate before they learn to talk. Manipulation, as a form of gesticulation, is often used when people speak to each other about some object. Naturalness of expression, non-encumbered interaction, intuitiveness and high sterility are all good reasons to replace the current interface technology (e.g., keyboard, mouse, and joystick) with more natural interfaces.

The operation of the gesture interface was tested at the Washington Hospital Center 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.

CHAPTER 2

OBJECTIVE

This paper presents a video-based hand gesture capture and recognition system used to manipulate magnetic resonance images (MRI) within a graphical user interface. A hand gesture vocabulary of commands was selected as being natural in the sense that each gesture is cognitively associated with the notion or command that is meant to represent it. For example, moving the hand left represents a “turn left” command.

CHAPTER 3: IDEATION PHASE

3.1 Literature Survey

Paper 1:

Bacterial contamination of computer keyboards in a teaching hospital.

Author : Schultz M, Gill J, Zubairi S, Huber R, Gordin F

https://pubmed.ncbi.nlm.nih.gov/12725363/

We tested 100 keyboards in 29 clinical areas for bacterial contamination. Ninety five were positive for microorganisms. Streptococcus, Clostridium perfringens, Enterococcus (including one vancomycin-resistant Enterococcus), Staphylococcus aureus, fungi, and gram-negative organisms were isolated.

Paper 2:

Face Mouse: A Novel Human-Machine Interface for Controlling the Position of a Laparoscope

Author : Nishikawa A, Hosoi T, Koara K, Negoro D, Hikita A, Asano S, Kakutani H, Miyazaki F

https://ieeexplore.ieee.org/document/1236756

Robotic laparoscope positioners are now expected as assisting devices for solo surgery among endoscopic surgeons. In such robotic systems, the human-machine (surgeon-robot) interface is of paramount importance because it is the means by which the surgeon communicates with and controls the robotic camera assistant. We have

designed a novel human-machine interface, called "FAce MOUSe", for controlling the position of a laparoscope.

Paper 3:

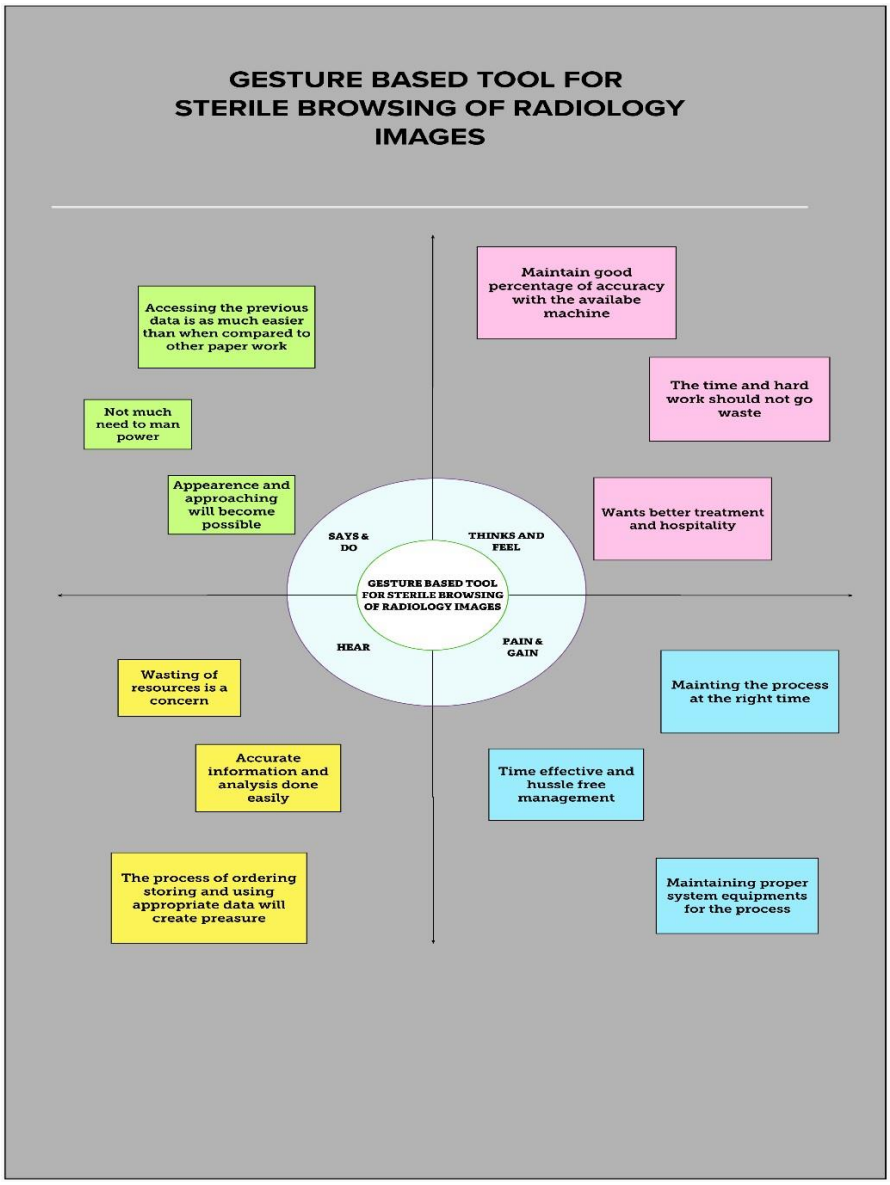
The NeuroStation- a highly accurate, minimally invasive solution to frameless stereotatic neurosurgery

Author : Smith KR, Frank KJ, Bucholz RD

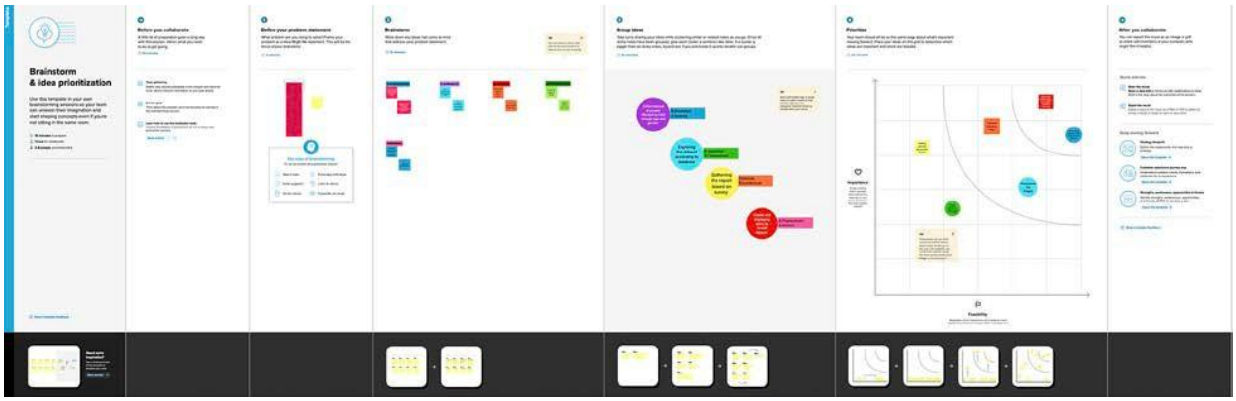
https://pubmed.ncbi.nlm.nih.gov/7923044/

The NeuroStation is an image-guided neurosurgery workstation designed to deliver frameless stereotaxy within an ergonomic, integrated surgical environment. Generally, stereotaxy can provide the neurosurgeon with important intra-operative localization information using diagnostic images such as computerized tomography (CT) or magnetic resonance imaging (MRI). To date, however, stereotaxy has not been widely accepted by neurosurgeons due to the procedural difficulties of incorporating conventional stereotaxy.

3.2 Empathy Map



3.2 Ideation



3.2 Problem Statement

Date	19 September 2022
Team ID	PNT2022TMID10032
Project Name	Project - A Gesture-based Tool for Sterile Browsing of Radiology Images
Maximum Marks	2 Marks
Who does the problem affect?	Different gestures can be predicted by this problem whose sterile images are uploaded and needed for the output.

What are the boundaries of the problem?	Several health conditions, your lifestyle, and your age and family history can increase your risk for heart disease.
What technology used to solve the issue?	Supervised and Un-supervised machine learning, Data mining, Computer vision with OpenCV, Python web application interface - Flask, Jupyter Notebook, IBM Cloud.

Why is it important that we fix the problem?	Predict if the patient suffers from different disease. The health professional enters the input values from the patient's health report. The data is fed into model which predicts different hand gestures based on the input values entered.

4. Project Design Phase-I

4.1 Proposed Solution - A Gesture-based Tool for Sterile Browsing of Radiology Images

Project Design Phase-I Proposed Solution

Date	02/11/2022
Team Id	PNT2022TMID10032
Project Name	Visualizing and Predicting Heart Diseases with an Interactive Dashboard
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To replicate sterile browsing skill in computers using image capture technology and classification techniques
2.	Idea / Solution description	A video based hand gesture capture and recognition system used to manipulate MRI within a graphical user interface

3.	Novelty / Uniqueness	By tracking the navigation and other gestures and translate to commands based on the temporal trajectories through video capture
4.	Social Impact / Customer Satisfaction	Doctors can analyse the image by having non-verbal communication
5.	Business Model (Revenue Model)	The business model of this system extracts intension and attention cues from the surgeon's behaviour. Hence, it is useful for the doctors and surgeons from any domain or region all over the world
6.	Scalability of the Solution	By adding few more gestures for manipulating the MRI images which are most essential for sterile browsing by doctors

4.2 Problem – Solution Fit

Date	05/11/2022
Team ID	PNT2022TMID10032
Project Name	Project - Visualizing and Predicting Heart Diseases with an Interactive Dashboard
Maximum Marks	2 Marks

Problem – Solution Fit Template:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioural patterns and recognize what would work and why.

Purpose:

- Solve complex problems in a way that fits the state of your customers.
- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behaviour.
- Sharpen your communication and marketing strategy with the right triggers and messaging.
- Increase touch-points with your company by finding the right problem-behaviour fit and building trust by solving frequent annoyances, or urgent or costly problems.

Understand the existing situation in order to improve it for your target group.

5. PROJECT DESIGN PHASE 2

Project Title: A Gesture-based Tool for Sterile Browsing of Radiology Image | **Project Design Phase-I - Solution Fit** | **Team ID PNT2022TMID10032**

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Neurology Doctors who analyzes radiology images.

6. CUSTOMER CC

As India is a developing country, the technology growth is still in progress. In medical field also the implementation of modern technology is growing. So, this concept is not aware among people in medical field and the cost is also relatively high to deploy.

5. AVAILABLE SOLUTIONS AS

A video-based hand gesture capture and recognition system used to manipulate MRI within a graphical user interface.
A hand gesture vocabulary of commands are selected such as 1-Resize, 2-Flip, 3-Rotate, 4-Rectangle.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Neurology doctors face problem of spreading infection while using computer keyboards and mouse in Intensive Care Unit(ICU) to view the radiology images.

9. PROBLEM ROOT CAUSE RC

The use of computer keyboards and mice by doctors and nurses in intensive care units (ICUs) is a common method for spreading infections.

7. BEHAVIOUR BE

The doctors/surgeons wish to browse the image database. The hand is moved rapidly out of the neural area so that a non-verbal interaction is possible to analyze the image.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

The use of hand gestures as an alternative to existing interface technologies offers the advantage of sterility.

10. YOUR SOLUTION SL

The image of the gesture captured in the video frame is compared with the pre-trained model and the gesture is identified. This

8. CHANNELS of BEHAVIOUR CH

Since this is viewing the images to analyze the details about the image both online and offline modes are possible.

Identify strong TR & EM

5.1 Customer Journey Map - A Gesture-based Tool for Sterile Browsing of Radiology Images

Date	07/11/2022
Team ID	PNT2022TMID10032
Project Name	Visualizing and Predicting Heart Diseases with an Interactive Dash Board

Customer Journey Map:

The customer journey map is a visual representation of the steps a customer takes to complete a specific action, such as signing up for a product trial or subscribing to a newsletter. The more steps involved to complete the specific action, the more detailed the customer journey map will be.

5.2 Solution Requirements

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	23/10/2022
Team ID	PNT2022TMID10032
Project Name	A Gesture-based Tool for Sterile Browsing of Radiology Images
Maximum Marks	4 Marks

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 through Gmail.
FR-2	User Confirmation	Confirmation via Email.
FR-3	User Sign up	Sign up using Gmail, user ID and password.
FR-4	User Login	Login using user ID and password.
FR-5	User Input	Setting unique hand gestures for easy and nonverbal communication.
FR-6	User Application	Input during usage of application to analyse the image via gestures.

Non-functional Requirements:

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

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User friendly as the instructions are displayed to the user when they enter the home page. The page will load in a short duration.
NFR-2	**Security**	The user can only login with their user ID and password. The data will be protected from the unauthorized user
NFR-3	**Reliability**	The application will perform up to 80% without failure (in predicting the gesture)
NFR-4	**Performance**	The application will respond within short duration provided the reasonable network speed.
NFR-5	**Availability**	The application will be available as a web page. Like other websites this can be accessed with the domain name. This is available as free service
NFR-6	**Scalability**	The application can be able to support the workload provided by the user to resize the image to their convenience. By adding few more gesturing for manipulation the MRI images which are most essential for sterile browsing by doctors

5.3 Data Flow Diagrams and User Stories - Visualizing and Predicting Heart Diseases with an Interactive Dash Board

Data Flow Diagrams:
Project Design Phase-II
Data Flow Diagram & User Stories

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.

Date	23/10/2022
Team ID	PNT2022TMID10032
Project Name	Project – A Gesture Based tool for sterile browsing of radiology images
Maximum Marks	4 marks

Flow:

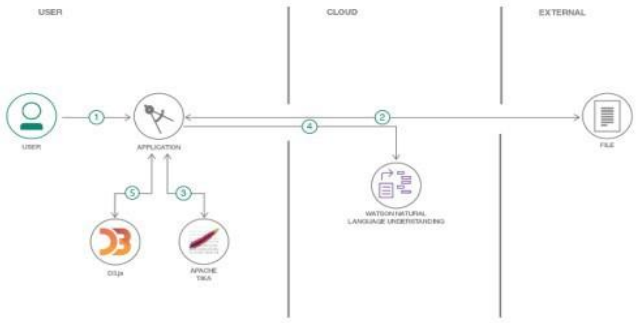
1) User creates an account in the application.
2) User enters the medical records in the dashboard.
3) User can view the visualizations of trends in the form of graphs and charts for his/her medical records with the trained dataset.
4) User can view the accuracy of probability of occurrence of heart disease in the dashboard.

Data Flow Diagram:

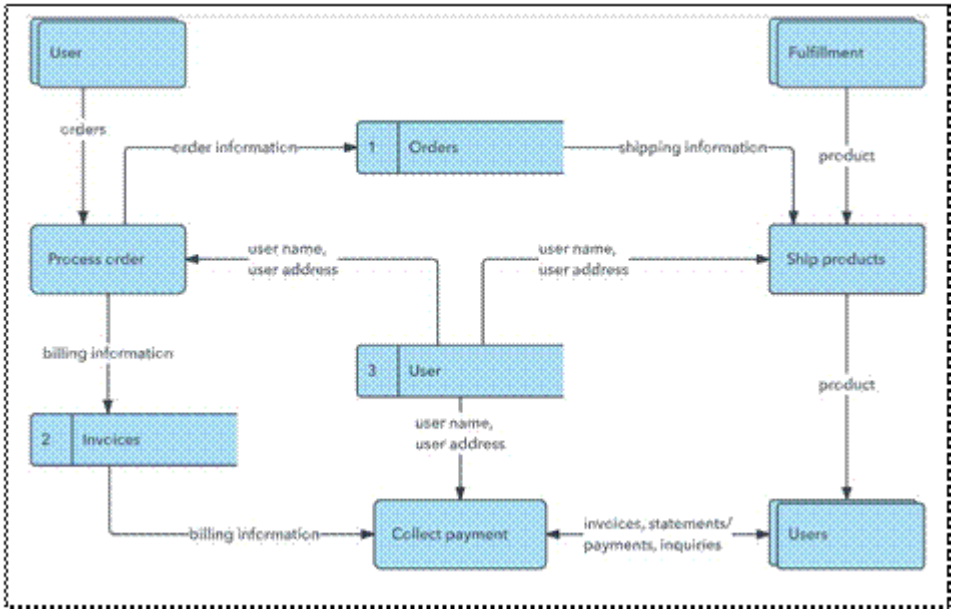
Date	23/10/2022
Team ID	PNT2022TMID10032
Project Name	A Gesture-based Tool for Sterile Browsing of Radiology Images
Maximum Marks	4 Marks

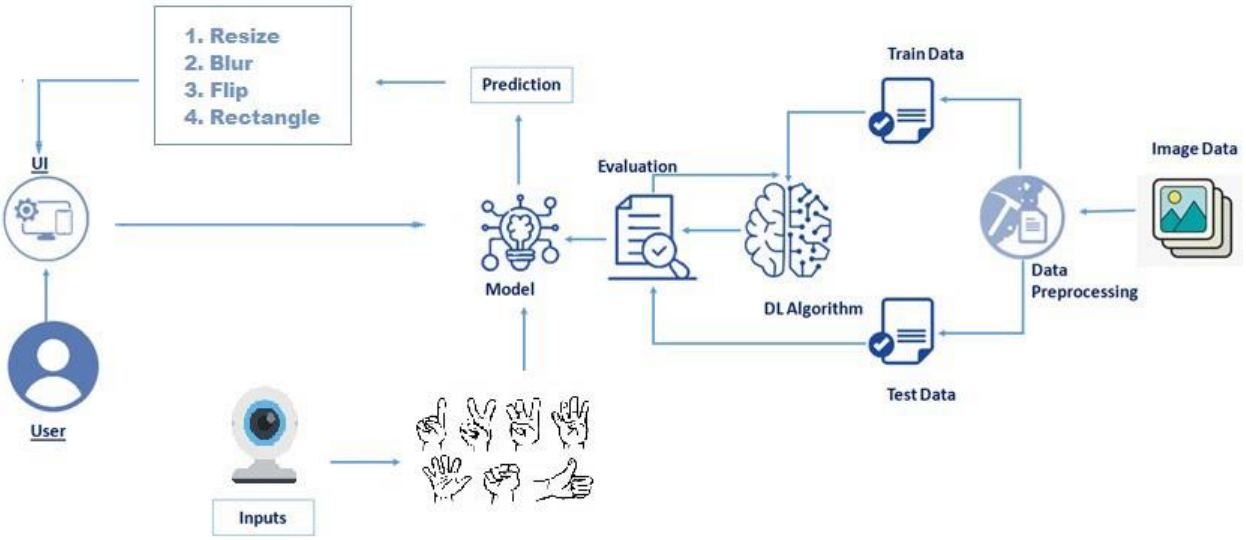
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored

Flow



1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.


User
Fulfillment
orders
order information
1 Orders
shipping information
product
Process order
user name,
user address
user name,
user address
Ship products
billing information
3 User
product
2 Invoices
user name,
user address
billing information
Collect payment
invoices, statements/
payments, inquiries
Users


1. Resize
2. Blur
3. Flip
4. Rectangle
Prediction
Train Data
UI
Evaluation
Image Data
Model
DL Algorithm
Data
Preprocessing
Test Data
User
Inputs

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Doctor)	Medical image manipulation	USN-1	As a user, I can make use of medical image manipulation and providing faster reponses at critical times	I can access the image manipulation data faster as before	High	Sprint-1
		USN-2	As a user, This model has best ease of use— the system allows me to use just my hands as	I can achieve the set target in short span of time with ease of use	High	Sprint-1

			a natural work tool			
Customer (Surgeon)	Gesture commands operation in real-time	USN-1	As a user, this prevents my focus shift and change of location while achieving a rapid intuitive reaction and easy interaction.	I can use the browsing of data with sterile postures	High	Sprint-1
		USN-2	As a user, this model responds to the surgeon's gesture commands in real-time (intuitive and fast)	I can access the manipulated images very fast and intuitive	High	Sprint-1

5.3 Technology Stack

Date	23/10/2022
Team ID	PNT2022TMID10032
Project Name	A Gesture-based Tool for Sterile Browsing of Radiology Images
Maximum Marks	4 Marks

The architectural diagram of the model is as below and the Technology used is shown in table1 & table 2

A Gesture- based tool for sterile browsing of Radiology Images

References:

https://www.researchgate.net/publication/351035037_Creating_domain_specific_chatbot_using_IBM_Watson

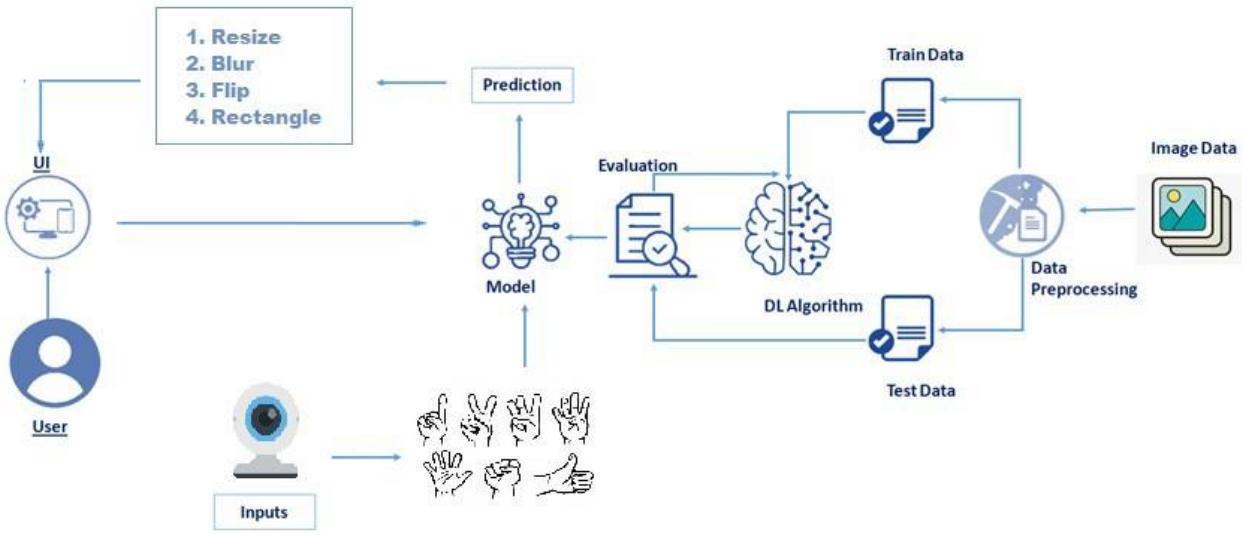


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI..	HTML, CSS, JavaScript.
2.	Application Logic-1	Upload image in an application	Python
3.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
4.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
5.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.
6.	Convolutional Neural Network	Initialize the model	CNN Layer

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Tensor flow,Theano, RNN, pyTorch, Flask
2.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Firewall and other security related softwares
3.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Data, models, operate at size, speed, consistency and complexity
4.	Performance	The system responds to the user in a second and the hardware and software works well	Image and facial recognition, speech recognition and real time captioning

References: **https://www.ibm.com/cloud/architecture**

6. PROJECT PLANNING

6.1 Milestone and Activity List

Date	23/10/2022
Team ID	PNT2022TMID10032
Project Name	A Gesture-based Tool for Sterile Browsing of Radiology Images

Activities:

S. No:	Milestone	Activities	Team Members
01.	Data Collection	Download the Dataset	Vignesh K Pugazhendhi N
02.	Data Collection	Image Pre-processing	Venkadesh R Praison Solomon V
03.	Data Collection	Import the Image Data Generator Library	Pugazhendhi N Venkadesh R
04.	Data Collection	Configure Image Data Generator Class	Vignesh K Praison Solomon V
05.	Data Collection	Apply Image Data Generator	Vignesh K Venkadesh R

		Functionality to Trainset and Test set	
06.	Model Building	Import the Model Building Libraries	Pugazhendhi N Praison Solomon V
07.	Model Building	Initializing the Model	Pugazhendhi N Praison Solomon V
08.	Model Building	Adding CNN Layers	Vignesh K Pugazhendhi N
09.	Model Building	Adding Dense Layers	Venkadesh R Praison Solomon V
10.	Model Building	Configure the Learning Process	Pugazhendhi N Venkadesh R
11.	Model Building	Train The Model	Vignesh K Venkadesh R
12.	Model Building	Save the Model	Vignesh K Praison Solomon V
13.	Model Building	Test Model	Vignesh K Pugazhendhi N Venkadesh R Praison Solomon V
14.	Application Building	Create HTML Pages	Vignesh K Pugazhendhi N Venkadesh R Praison Solomon V
15.	Application Building	Build Python code	Vignesh K Pugazhendhi N Venkadesh R

			Praison Solomon V
16.	Application Building	Run the Application	Vignesh K Pugazhendhi N Venkadesh R Praison Solomon V
17.	Train The Model on IBM	Register for IBM Cloud	Vignesh K Pugazhendhi N Venkadesh R Praison Solomon V
18.	Train The Model on IBM	Train Model on IBM	Vignesh K Pugazhendhi N Venkadesh R Praison Solomon V

6.2 Sprint Delivery Plan

Date	23/10/2022
Team ID	PNT2022TMID10032
Project Name	A Gesture-based Tool for Sterile Browsing of Radiology Images
Maximum marks	8 Marks

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	Venkadesh R Praison Solomon V
Sprint-1		USN-2	Image Pre-processing	10	High	Vignesh K Pugazhendhi N
Sprint-1		USN-3	Import and Configure the Image Data Generator Library and Class	10	High	Venkadesh R Praison Solomon V
Sprint-1		USN-4	Apply Image Data Generator	10	High	Venkadesh R Praison Solomon V

			Functionality to Train-Set and Test-Set			
Sprint -2	**Model Building**	USN-5	Import the Model Building Libraries and Initializing the Model	10	High	Vignesh K Pugazhendhi N
Sprint -2		USN-6	Adding CNN Layers and Dense Layers	10	High	Venkadesh R Vignesh K
Sprint -2		USN-7	Configure the Learning Process	10	High	Pugazhendhi N Praison Solomon V
Sprint -2		USN-8	Train the Model, Save the Model and Test the Model	10	High	Praison Solomon V Pugazhendhi N
Sprint -3	**Application Building**	USN-9	Create Web Application using HTML, CSS, JavaScript	10	High	Venkadesh R Praison Solomon V
Sprint -3		USN-10	Build Python code	10	High	Venkadesh R Praison Solomon V

Sprint -4	**Train The Model on IBM**	USN-11	Register for IBM Cloud	10	High	Vignesh K Pugazhendhi N
Sprint -4		USN-12	Train the Model and Test the Model and its Overall Performance	10	High	Pugazhendhi N Venkadesh R

Project Tracker, Velocity & Burndown Chart: (4 Marks)

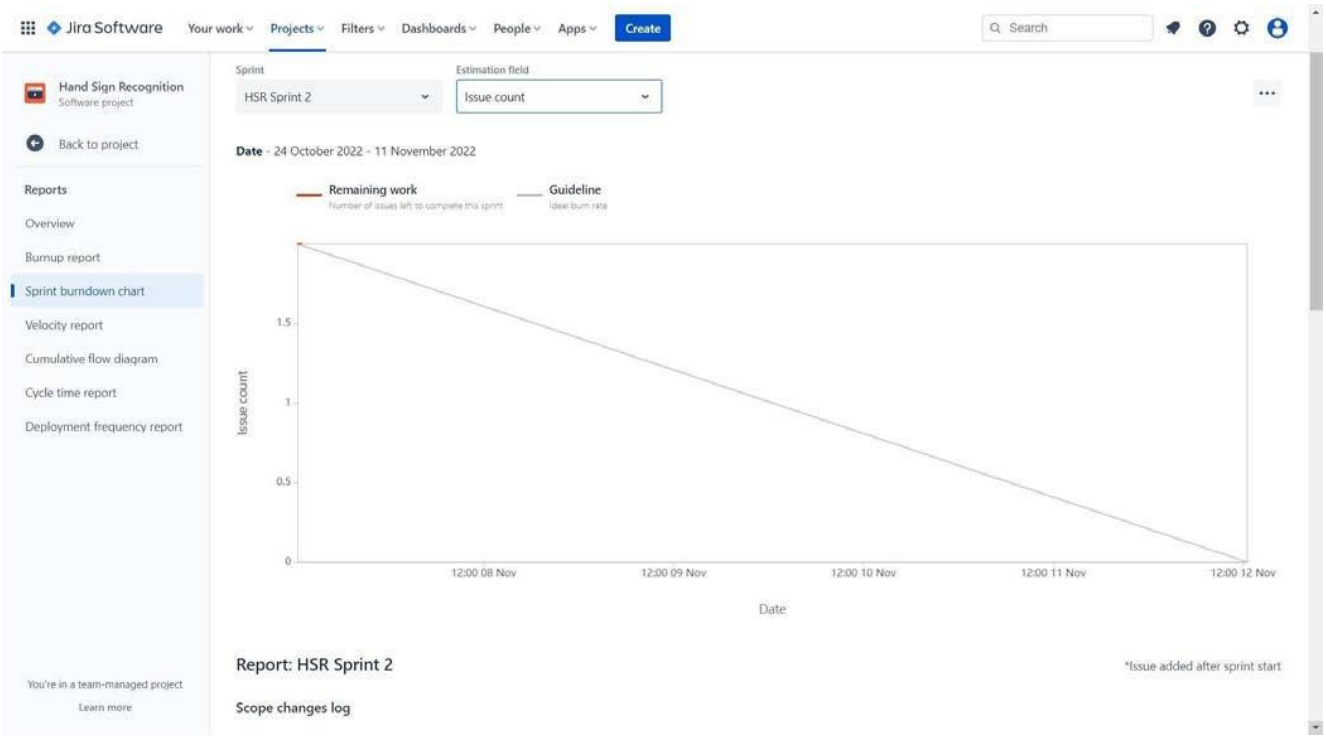
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Sprint	Total Score 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

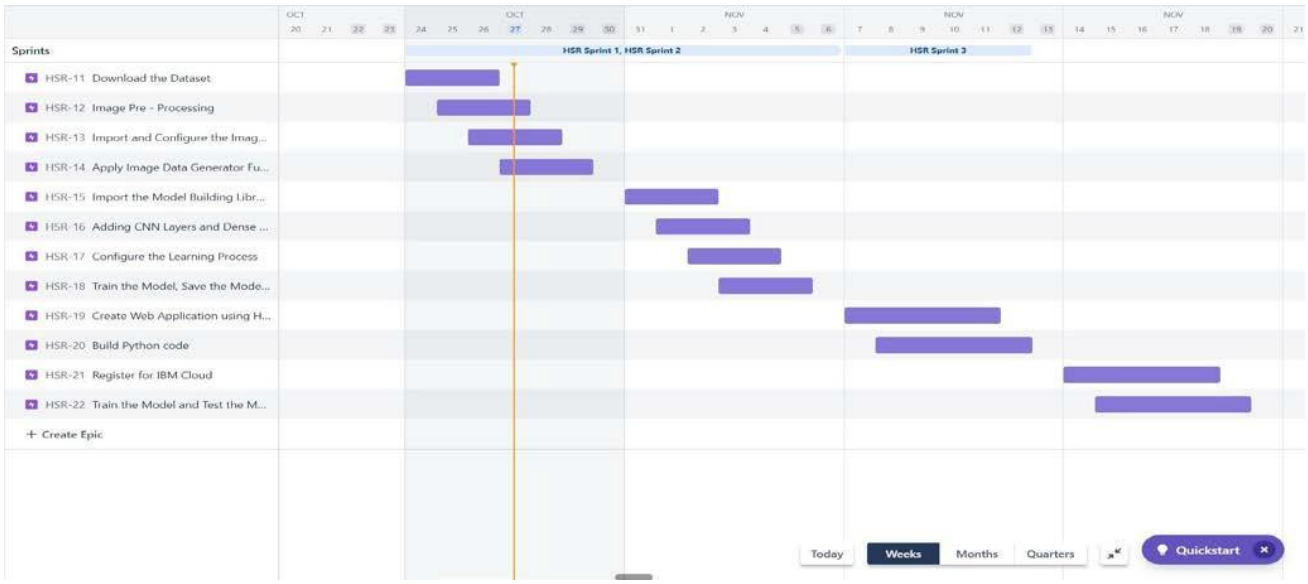
Velocity:

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) .

Burndown Chart:



Road Map:


OCT
OCT
NOV
NOV
NOV
Sprints
HSR Sprint 1, HSR Sprint 2
HSR Sprint 3
HSR-11 Download the Dataset
HSR-12 Image Pre - Processing
HSR-13 Import and Configure the Imag...
HSR-14 Apply Image Data Generator Fu...
HSR-15 Import the Model Building Libr...
HSR-16 Adding CNN Layers and Dense ...
HSR-17 Configure the Learning Process
HSR-18 Train the Model, Save the Mode...
HSR-19 Create Web Application using H...
HSR-20 Build Python code
HSR-21 Register for IBM Cloud
HSR-22 Train the Model and Test the M...
+ Create Epic
Today
Weeks
Months
Quarters
Quickstart

7 PROJECT DEVELOPMENT PHASE

7.1 Project Development - Delivery of Sprint - 1

Image Pre-processing

Team ID : PNT2022TMID10032

Project Title : A Gesture-based Tool for Sterile Browsing of Radiology Images.

IMAGE PREPROCESSING

In [40]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

In [41]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [42]:
```
train_data_generator1 = ImageDataGenerator(rescale=1.0/255,horizontal_flip=True)
test_data_generator1 = ImageDataGenerator(rescale=1.0/255, horizontal_flip=True)
```

In [43]:
```
train_data_generator2 = ImageDataGenerator(rescale=1.0/255,rotation_range=90)
test_data_generator2 = ImageDataGenerator(rescale=1.0/255,rotation_range=90)
```

In [46]:
```
train_data_generator3 = ImageDataGenerator(rescale=1.0/255,brightness_range=[0.2,1.0])
test_data_generator3 = ImageDataGenerator(rescale=1.0/255,brightness_range=[0.2,1.0])
train_data_generator4 = ImageDataGenerator(rescale=1.0/255,zoom_range=[0.5,1.0])
test_data_generator4 = ImageDataGenerator(rescale=1.0/255,zoom_range=[0.5,1.0])
```

In [47]:
```
trdata1 = train_data_generator1.flow_from_directory('/content/drive/MyDrive/train-20221106T023729Z-001/train',target_size=(64,64),batch_size=(3),color_mo
```

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

In [48]:
```
trdata2 = train_data_generator2.flow_from_directory('/content/drive/MyDrive/train-20221106T023729Z-001/train',target_size=(64,64),batch_size=(3),color_mo
```

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

In [49]:
```
trdata3 = train_data_generator3.flow_from_directory('/content/drive/MyDrive/train-20221106T023729Z-001/train',target_size=(64,64),batch_size=(3),color_mo
```

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

In [50]:
```
trdata4 = train_data_generator4.flow_from_directory('/content/drive/MyDrive/train-20221106T023729Z-001/train',target_size=(64,64),batch_size=(3),color_mo
```

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

In [51]:
```
tsdata1 = test_data_generator1.flow_from_directory('/content/drive/MyDrive/train-20221106T023729Z-001/train',target_size=(64,64),batch_size=32,class_mode
```

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

In [52]:
```
tsdata2 = test_data_generator2.flow_from_directory('/content/drive/MyDrive/test-20221106T023808Z-001/test',target_size=(64,64),batch_size=32,class_mode='
```

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

In [53]:
```
tsdata3 = test_data_generator3.flow_from_directory('/content/drive/MyDrive/test-20221106T023808Z-001/test',target_size=(64,64),batch_size=32,class_mode='
```

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

In [54]:
```
tsdata4 = test_data_generator4.flow_from_directory('/content/drive/MyDrive/test-20221106T023808Z-001/test',target_size=(64,64),batch_size=32,class_mode='
```

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

In [56]:
```
print(trdata2.class_indices)
```

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

In [57]:
```
print(trdata3.class_indices)
```

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

In [58]:
```
print(trdata4.class_indices)
```

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

In [59]:
```
print(tsdata1.class_indices)
```

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

In [61]:
```
print(tsdata2.class_indices)
```

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

In [62]:
```
print(tsdata3.class_indices)
```

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

In [63]:
```
print(tsdata4.class_indices)
```

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

Model Testing

Team ID : PNT2022TMID10032

Project Title : A Gesture-based Tool for Sterile Browsing of Radiology Images.

Model Testing

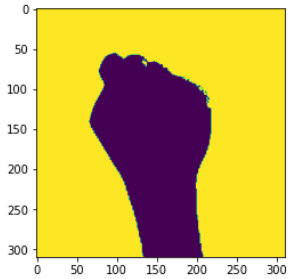
In [28]:
```
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model = load_model("gesture.h5")

path = r'/content/drive/MyDrive/test-20221106T023808Z-001/test/0/0.jpg'
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
x_train = train_datagen.flow_from_directory(r'/content/drive/MyDrive/train-20221106T023729Z-001/train',
                                            target_size=(64, 64),
                                            batch_size=3,
                                            color_mode='grayscale',
                                            class_mode='categorical')
x_test = test_datagen.flow_from_directory(r'/content/drive/MyDrive/test-20221106T023808Z-001/test',
                                          target_size=(64, 64),
                                          batch_size=3,
                                          color_mode='grayscale',
                                          class_mode='categorical')
```

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

In [29]:
```
%pylab inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
imgs = mpimg.imread(path)
imgplot = plt.imshow(imgs)
plt.show()
```

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



In [31]:
```
#loading of the image
img = image.load_img(path,
                     color_mode='grayscale',
                     target_size= (64,64))
x = image.img_to_array(img)#image to array
x.shape
```

Out[31]: (64, 64, 1)

In [32]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

In [33]:
```
type(x)
```

Out[33]:
```
numpy.ndarray
```

In [34]:
```
#changing the shape
x = np.expand_dims(x,axis = 0)
```

In [35]:
```
x.shape
```

Out[35]:
```
(1, 64, 64, 1)
```

In [36]:
```
pred = model.predict(x)
pred
```

```
1/1 [==============================] - 0s 90ms/step
```

Out[36]:
```
array([[1., 0., 0., 0., 0., 0.]], dtype=float32)
```

In [37]:
```
x_test.class_indices
```

Out[37]:
```
{'0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5}
```

In [38]:
```
index=['0','1','2','3','4','5']
```

MODEL BUILDING

Team ID : PNT2022TMID10032

Project Title : A Gesture-based Tool for Sterile Browsing of Radiology Images.

Model Training

Importing Packages

In [89]:
```
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 [5]:
```
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 [75]:
```
x_test = test_datagen.flow_from_directory(r'/content/drive/MyDrive/train-20221106T023729Z-001/train', target_size=(64, 64), batch_size=3, color_mode='gra
x_train = train_datagen.flow_from_directory(r'/content/drive/MyDrive/test-20221106T023808Z-001/test', target_size=(64, 64), batch_size=3, color_mode='gra
```

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

In [76]:
```
print(x_train.class_indices)
```

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

Initializing The Model

In [77]:
```
model=Sequential()
```

Adding CNN Layers

In [78]:
```
model.add(Conv2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

In [79]:
```
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

In [80]:
```
model.add(Flatten())
```

Adding Dense Layers

In [81]:
```
model.add(Dense(units=512, activation='relu'))
```

In [82]:
```
model.add(Dense(units=6, activation='softmax'))
```

In [83]:
```
model.summary()
```

In [90]:
```
model.fit_generator(x_train,steps_per_epoch= len(x_train),
                    epochs= 25,
                    validation_data=x_test,
                    validation_steps=len(x_train))
```

```
Epoch 1/25
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
  after removing the cwd from sys.path.
10/10 [==============================] - 2s 108ms/step - loss: 0.0838 - accuracy: 0.9667 - val_loss: 4.8002 - val_accuracy: 0.3667
Epoch 2/25
10/10 [==============================] - 1s 88ms/step - loss: 0.0481 - accuracy: 1.0000 - val_loss: 3.2492 - val_accuracy: 0.5667
Epoch 3/25
10/10 [==============================] - 1s 90ms/step - loss: 0.1411 - accuracy: 0.9333 - val_loss: 2.1161 - val_accuracy: 0.6000
Epoch 4/25
10/10 [==============================] - 1s 102ms/step - loss: 0.0870 - accuracy: 0.9667 - val_loss: 3.3258 - val_accuracy: 0.3667
Epoch 5/25
10/10 [==============================] - 1s 86ms/step - loss: 0.0609 - accuracy: 0.9667 - val_loss: 4.7341 - val_accuracy: 0.3667
Epoch 6/25
10/10 [==============================] - 1s 102ms/step - loss: 0.0070 - accuracy: 1.0000 - val_loss: 2.9502 - val_accuracy: 0.5667
Epoch 7/25
10/10 [==============================] - 1s 87ms/step - loss: 0.0352 - accuracy: 0.9667 - val_loss: 2.4555 - val_accuracy: 0.6333
Epoch 8/25
10/10 [==============================] - 1s 92ms/step - loss: 0.0440 - accuracy: 0.9667 - val_loss: 3.2904 - val_accuracy: 0.5667
Epoch 9/25
10/10 [==============================] - 1s 106ms/step - loss: 0.0055 - accuracy: 1.0000 - val_loss: 2.6798 - val_accuracy: 0.6667
Epoch 10/25
10/10 [==============================] - 1s 88ms/step - loss: 0.0054 - accuracy: 1.0000 - val_loss: 4.6708 - val_accuracy: 0.4333
Epoch 11/25
10/10 [==============================] - 1s 86ms/step - loss: 0.0044 - accuracy: 1.0000 - val_loss: 4.1461 - val_accuracy: 0.5333
Epoch 12/25
10/10 [==============================] - 1s 91ms/step - loss: 0.0031 - accuracy: 1.0000 - val_loss: 5.3703 - val_accuracy: 0.5667
Epoch 13/25
10/10 [==============================] - 1s 107ms/step - loss: 0.0010 - accuracy: 1.0000 - val_loss: 3.1483 - val_accuracy: 0.6000
Epoch 14/25
```

```
10/10 [==============================] - 1s 91ms/step - loss: 0.0031 - accuracy: 1.0000 - val_loss: 5.3703 - val_accuracy: 0.5667
Epoch 13/25
10/10 [==============================] - 1s 107ms/step - loss: 0.0010 - accuracy: 1.0000 - val_loss: 3.1483 - val_accuracy: 0.6000
Epoch 14/25
10/10 [==============================] - 1s 92ms/step - loss: 0.0065 - accuracy: 1.0000 - val_loss: 3.0748 - val_accuracy: 0.5667
Epoch 15/25
10/10 [==============================] - 1s 106ms/step - loss: 0.0016 - accuracy: 1.0000 - val_loss: 4.2850 - val_accuracy: 0.7000
Epoch 16/25
10/10 [==============================] - 1s 105ms/step - loss: 6.9349e-04 - accuracy: 1.0000 - val_loss: 6.7626 - val_accuracy: 0.5333
Epoch 17/25
10/10 [==============================] - 1s 88ms/step - loss: 4.1833e-04 - accuracy: 1.0000 - val_loss: 4.0649 - val_accuracy: 0.4333
Epoch 18/25
10/10 [==============================] - 1s 89ms/step - loss: 2.2985e-04 - accuracy: 1.0000 - val_loss: 6.8380 - val_accuracy: 0.5000
Epoch 19/25
10/10 [==============================] - 1s 104ms/step - loss: 6.7672e-04 - accuracy: 1.0000 - val_loss: 5.0654 - val_accuracy: 0.5667
Epoch 20/25
10/10 [==============================] - 1s 89ms/step - loss: 8.1685e-04 - accuracy: 1.0000 - val_loss: 2.6641 - val_accuracy: 0.6667
Epoch 21/25
10/10 [==============================] - 1s 101ms/step - loss: 0.0016 - accuracy: 1.0000 - val_loss: 6.2709 - val_accuracy: 0.5333
Epoch 22/25
10/10 [==============================] - 1s 92ms/step - loss: 3.6468e-04 - accuracy: 1.0000 - val_loss: 6.2225 - val_accuracy: 0.5333
Epoch 23/25
10/10 [==============================] - 1s 87ms/step - loss: 2.6394e-04 - accuracy: 1.0000 - val_loss: 2.2229 - val_accuracy: 0.7333
Epoch 24/25
10/10 [==============================] - 1s 88ms/step - loss: 1.9120e-04 - accuracy: 1.0000 - val_loss: 7.6336 - val_accuracy: 0.5333
Epoch 25/25
10/10 [==============================] - 1s 88ms/step - loss: 1.4391e-04 - accuracy: 1.0000 - val_loss: 4.6080 - val_accuracy: 0.5000
```

Out[90]:

Save The Model

In [91]:
```
model.save('gesture.h5')
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 62, 62, 32)        320

 max_pooling2d_4 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_5 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_2 (Flatten)         (None, 6272)              0

 dense_10 (Dense)            (None, 512)               3211776

 dense_11 (Dense)            (None, 6)                 3078

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

Configure The Learning Process

In [88]:
```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Train The Model

7.2 Project Development - Delivery of Sprint - 2

Templates:

Home:

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
   <meta name="description" content="Start your development with Creative Design
landing page.">
   <meta name="author" content="Devcrud">
   <title>Hand Gesture Recognition</title>
   <link rel="stylesheet" href="../static/vendors/themify-icons/css/themify-icons.css">
        <link rel="stylesheet" href="../static/css/creative-design.css">
<style>
.header {
  background: #efefef url(../static/imgs/3.jpg);
  background-size: cover;
  background-position: center center;
  background-repeat: no-repeat;
  text-align: center;
  color: white;
  position: relative;
  height: 598px;
  position: relative;
}
</style>
</head>
<body data-spy="scroll" data-target=".navbar" data-offset="40" id="home">
   <!-- Page Navbar -->
   <nav id="scrollspy" class="navbar page-navbar navbar-light navbar-expand-md fixed-
top" data-spy="affix" data-offset-top="20">
      <div class="container">
```

```
        <a class="navbar-brand" href="#"><strong class="text-primary">Hand</strong>
<span class="text-dark">Gesture</span></a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item">
                    <a class="nav-link" href="home.html">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="intro.html">Introduction</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="index6.html">Predict</a>
                </li>
            </ul>
        </div>
    </div>
  </nav><!-- End of Page Navbar -->

  <!-- Page Header -->
  <header id="home" class="header">
<img src="../static/imgs/11.png" style="width:1000px;height:600px;">

    <div class="overlay"></div>
    <div class="header-content">
        <p>Image Processing Using Hand Gesture</p>
        <h1>A Gesture-based Tool for Sterile Browsing of Radiology Images</h1>
    </div>
  </header><!-- End of Page Header -->
```

Index:

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
   <meta name="description" content="Start your development with Creative Design
landing page.">
   <meta name="author" content="Devcrud">
   <title>Hand Gesture Recognition</title>
   <link rel="stylesheet" href="../static/vendors/themify-icons/css/themify-icons.css">
        <link rel="stylesheet" href="../static/css/creative-design.css">
<style>
.header {
  background: #efefef url(../static/imgs/3.jpg);
  background-size: cover;
  background-position: center center;
  background-repeat: no-repeat;
  text-align: center;
  color: white;
  position: relative;
  height: 598px;
  position: relative;
}
</style>
</head>
<body data-spy="scroll" data-target=".navbar" data-offset="40" id="home">
   <!-- Page Navbar -->
   <nav id="scrollspy" class="navbar page-navbar navbar-light navbar-expand-md fixed-
top" data-spy="affix" data-offset-top="20">
      <div class="container">
         <a class="navbar-brand" href="#"><strong class="text-primary">Hand</strong>
<span class="text-dark">Gesture</span></a>
         <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
```

```
        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="home.html">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="intro.html">Introduction</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="index6.html">Predict</a>
            </li>
          </ul>
        </div>
      </div>
    </nav><!-- End of Page Navbar -->

    <!-- Page Header -->
    <header id="home" class="header">
<img src="../static/imgs/11.png" style="width:1000px;height:600px;">

      <div class="overlay"></div>
      <div class="header-content">
        <p>Image Processing Using Hand Gesture</p>
        <h1>A Gesture-based Tool for Sterile Browsing of Radiology Images</h1>
        <button class="btn btn-outline-light">Upload Image</button>
      </div>
    </header><!-- End of Page Header -->
```

Intro:

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
```

```
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
   <meta name="description" content="Start your development with Creative Design
landing page.">
   <meta name="author" content="Devcrud">
   <title>Hand Gesture Recognition</title>
   <link rel="stylesheet" href="../static/vendors/themify-icons/css/themify-icons.css">
         <link rel="stylesheet" href="../static/css/creative-design.css">
<style>
.header {
  background: #efefef url(../static/imgs/1.jpg);
  background-size: cover;
  background-position: center center;
  background-repeat: no-repeat;
  text-align: center;
  color: white;
  position: relative;
  height: 598px;
  position: relative;
}
</style>
</head>
<body data-spy="scroll" data-target=".navbar" data-offset="40" id="home">
   <!-- Page Navbar -->
   <nav id="scrollspy" class="navbar page-navbar navbar-light navbar-expand-md fixed-
top" data-spy="affix" data-offset-top="20">
      <div class="container">
         <a class="navbar-brand" href="#"><strong class="text-primary">Hand</strong>
<span class="text-dark">Gesture</span></a>

         <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ml-auto">
               <li class="nav-item">
                  <a class="nav-link" href="home.html">Home</a>
               </li>
               <li class="nav-item">
                  <a class="nav-link" href="intro.html">Introduction</a>
               </li>
               <li class="nav-item">
                  <a class="nav-link" href="index6.html">Predict</a>
```

```
                </li>
              </ul>
            </div>
        </div>
    </nav><!-- End of Page Navbar -->

    <!-- Page Header -->
    <header id="home" class="header">

        <div class="overlay"></div>
        <div class="header-content">
            <p>Image Processing Using Hand Gesture</p>
            <h3 style="color:yellow;">A Gesture-based Tool for Sterile Browsing of
Radiology Images</h3>
                                        <h4>        Hand Gesture recognition system provides us
with an innovative, natural, user-friendly way of interacting with the computer which
is more familiar to human beings. In our project, the hand region is extracted
from the background by using the Region of interest. Then, we will be predicted
the labels based on the CNN-trained model weights of hand gestures using that predicted
labels we apply if conditions to control some of the actions like reshaping, blurring, and
flip of the given image. <h4>
        </div>
    </header><!-- End of Page Header -->

    <!DOCTYPE html>
<html lang="en">
<head>
            <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <meta name="description" content="Start your development with Creative Design
landing page.">
    <meta name="author" content="Devcrud">
    <title>Hand Gesture Recognition</title>
    <link rel="stylesheet" href="../static/vendors/themify-icons/css/themify-icons.css">
            <link rel="stylesheet" href="../static/css/creative-design.css">
<style>
.header {
  background: #efefef url(../static/imgs/1.jpg);
  background-size: cover;
```

```
  background-position: center center;
  background-repeat: no-repeat;
  text-align: center;
  color: white;
  position: relative;
  height: 598px;
  position: relative;
}
</style>
</head>
<body data-spy="scroll" data-target=".navbar" data-offset="40" id="home">
    <!-- Page Navbar -->
    <nav id="scrollspy" class="navbar page-navbar navbar-light navbar-expand-md fixed-
top" data-spy="affix" data-offset-top="20">
        <div class="container">
            <a class="navbar-brand" href="#"><strong class="text-primary">Hand</strong>
<span class="text-dark">Gesture</span></a>

            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav ml-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="home.html">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="intro.html">Introduction</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="index6.html">Predict</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav><!-- End of Page Navbar -->

    <!-- Page Header -->
    <header id="home" class="header">

        <div class="overlay"></div>
        <div class="header-content">
            <p>Image Processing Using Hand Gesture</p>
```

<h3 style="color:yellow;">A Gesture-based Tool for Sterile Browsing of Radiology Images</h3>

<h4> Hand Gesture recognition system provides us with an innovative, natural, user-friendly way of interacting with the computer which is more familiar to human beings. In our project, the hand region is extracted from the background by using the Region of interest. Then, we will be predicted the labels based on the CNN-trained model weights of hand gestures using that predicted labels we apply if conditions to control some of the actions like reshaping, blurring, and flip of the given image. <h4>

</header><!-- End of Page Header -->

7.3 Project Development - Delivery of Sprint – 3

Build Python Code

- Build flask file ‘app.py’ which is a web framework written in pythonfor server-side scripting.
- App starts running when “ name ” constructor is called in main.

- render_template is used to return html file.
- “GET” method is used to take input from the user.
- “POST” method is used to display the output to the user.

- Importing Libraries

```
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
```

Run the Application:

At last, we will run our flask application

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

Run The app in local browser

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type “python app.py” command

Navigate to the localhost where you can view your web page

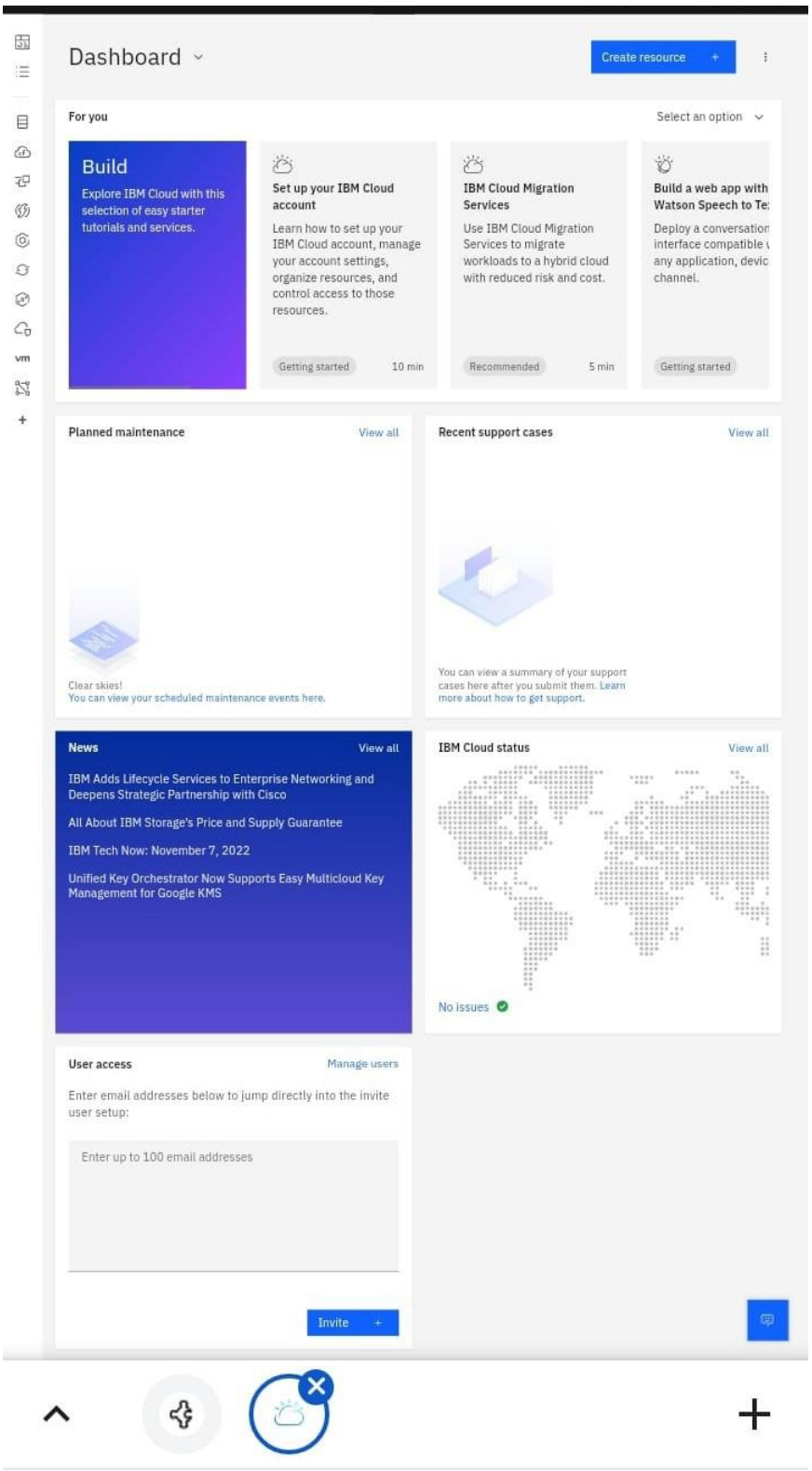
```
(base) E:\>cd E:\PROJECTS\number-sign-recognition\Flask

(base) E:\PROJECTS\number-sign-recognition\Flask>python app.py
```

Then it will run on localhost:5000

```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

7.4 Project Development - Delivery of Sprint - 4



Train Model On IBM:

```
import os, types
from ibm_watson_machine_learning import APIClient
import pandas as pd
from botocore.client import Config
import ibm_boto3
from io import BytesIO
import zipfile

#Due to privacy concerns, I've not mentioned the API Keys and Endpoints Here
def__iter__(self): return 0

cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id=<api_key>,
    ibm_auth_endpoint=<end_point>,
    config=Config(signature_version='oauth'),
    endpoint_url=<end_point_url>')

bucket = <bucket_name>
object_key = <object_key>

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
filepaths=unzip.namelist()
for path in filepaths:
    unzip.extract(path)

train_datagen = ImageDataGenerator( rescale=1./255,
                                    rotation_range=10.,
                                    width_shift_range=0.1,
                                    height_shift_range=0.1,
                                    zoom_range=0.2,
                                    horizontal_flip=True
                                   )
train_gen = train_datagen.flow_from_directory(
        r'/home/wsuser/work/Finger  Dataset/train',
        target_size=(128,128),
```

```
        color_mode='grayscale',
        batch_size=32,
        classes=['0','1','2','3','4','5'],
        class_mode='categorical'
    )
test_datagen = ImageDataGenerator( rescale=1./255 )
test_gen = test_datagen.flow_from_directory(
        r'/home/wsuser/work/Finger Dataset/test',
        target_size=(128,128),
        color_mode='grayscale',
        batch_size=32,
        classes=['0','1','2','3','4','5'],
        class_mode='categorical'
    )
model=Sequential()
model.add(BatchNormalization(input_shape = (128,128,1)))
model.add(Convolution2D(32, (3,3), activation ='relu', input_shape = (128, 128, 1)))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=6,kernel_size=4,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=3,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=2,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Flatten())
model.add(Dense(units=128,activation = 'relu'))
model.add(Dense(units = 64, activation = 'relu'))
model.add(Dense(units = 32, activation = 'relu'))
model.add(Dense(units = 6, activation = 'softmax'))
model.summary()
model.compile(optimizer='adam', loss = 'categorical_crossentropy',metrics = ['accuracy'])
model.fit_generator(train_gen,
                    epochs=20,
                    steps_per_epoch=18000//32,
                    validation_data=test_gen,
                    verbose = 1,validation_steps=3600//32)
model.save('gesture.h5')
wml_credentials={
    "url":'https://us-south.ml.cloud.ibm.com',
    "apikey":'on6wVLLy-ERS74JlvyDrFdJ35GRaHzaCtKxejqR7euwG'
```

```
}
client=APIClient(wml_credentials)


def guid_from_space_name(client,space_name):
   space=client.spaces.get_details()
   return(next(item for item in space['resources'] if
item['entity']['name']==space_name)['metadata']['id'])


space_uid=guid_from_space_name(client,'Gesture_Deploy')
client.set.default_space(space_uid)

software_spec_uid=client.software_specifications.get_uid_by_name('tensorflow_rt22.1-
py3.9')

!tar -zcvf gesture_based_tool.tgz gesture.h5

model_details=client.repository.store_model(model='gesture_based_tool.tgz',meta_props
={
                                client.repository.ModelMetaNames.NAME:"Gesture Based
Tool",
                                client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",

client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid
                                }
                               )
model_id=client.repository.get_model_id(model_details)
```

Footer

Footer navigation

Terms

Privacy

Security

Status

Docs

Contact GitHub

Pricing

API

Training

Blog
About

```
import os, types
from ibm_watson_machine_learning import APIClient
import pandas as pd
from botocore.client import Config
import ibm_boto3
from io import BytesIO
import zipfile

#Due to privacy concerns, I've not mentioned the API Keys and Endpoints Here
def__iter__(self): return 0

cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id=<api_key>,
    ibm_auth_endpoint=<end_point>,
    config=Config(signature_version='oauth'),
    endpoint_url=<end_point_url>')

bucket = <bucket_name>
object_key = <object_key>

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
filepaths=unzip.namelist()
for path in filepaths:
    unzip.extract(path)

train_datagen = ImageDataGenerator( rescale=1./255,
                                    rotation_range=10.,
                                    width_shift_range=0.1,
                                    height_shift_range=0.1,
                                    zoom_range=0.2,
                                    horizontal_flip=True
                                   )
train_gen = train_datagen.flow_from_directory(
        r'/home/wsuser/work/Finger Dataset/train',
        target_size=(128,128),
        color_mode='grayscale',
```

```
        batch_size=32,
        classes=['0','1','2','3','4','5'],
        class_mode='categorical'
    )
test_datagen = ImageDataGenerator( rescale=1./255 )
test_gen = test_datagen.flow_from_directory(
        r'/home/wsuser/work/Finger Dataset/test',
        target_size=(128,128),
        color_mode='grayscale',
        batch_size=32,
        classes=['0','1','2','3','4','5'],
        class_mode='categorical'
    )
model=Sequential()
model.add(BatchNormalization(input_shape = (128,128,1)))
model.add(Convolution2D(32, (3,3), activation ='relu', input_shape = (128, 128, 1)))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=6,kernel_size=4,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=3,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=2,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Flatten())
model.add(Dense(units=128,activation = 'relu'))
model.add(Dense(units = 64, activation = 'relu'))
model.add(Dense(units = 32, activation = 'relu'))
model.add(Dense(units = 6, activation = 'softmax'))
model.summary()
model.compile(optimizer='adam', loss = 'categorical_crossentropy',metrics = ['accuracy'])
model.fit_generator(train_gen,
                    epochs=20,
                    steps_per_epoch=18000//32,
                    validation_data=test_gen,
                    verbose = 1,validation_steps=3600//32)
model.save('gesture.h5')
wml_credentials={
   "url":'https://us-south.ml.cloud.ibm.com',
   "apikey":'on6wVLLy-ERS74JlvyDrFdJ35GRaHzaCtKxejqR7euwG'
}
```

```
client=APIClient(wml_credentials)


def guid_from_space_name(client,space_name):
   space=client.spaces.get_details()
   return(next(item for item in space['resources'] if
item['entity']['name']==space_name)['metadata']['id'])


space_uid=guid_from_space_name(client,'Gesture_Deploy')
client.set.default_space(space_uid)

software_spec_uid=client.software_specifications.get_uid_by_name('tensorflow_rt22.1-
py3.9')

!tar -zcvf gesture_based_tool.tgz gesture.h5

model_details=client.repository.store_model(model='gesture_based_tool.tgz',meta_props
={
                                client.repository.ModelMetaNames.NAME:"Gesture Based
Tool",
                                client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",

client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid
                                 }
                               )
model_id=client.repository.get_model_id(model_details)
```

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.

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. [PubMed] [Google Scholar]

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. [Google Scholar]

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. [PubMed] [Google Scholar]

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. [PubMed] [Google Scholar]

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. [Google Scholar]

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. [Google Scholar]

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

Demonstration Link: https://youtu.be/Iiu90IDvp7U

Github Repo Link: https://github.com/IBM-EPBL/IBM-Project-36502-1660295611