A GESTURE BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

By:

Team ID: PNT2022TMID03497

APPIREDDIGARI NITHIN KUMAR REDDY (212219060022)

PAPPU VENKATA TEJA (212219060193)

ANUMALA CHANDU (212219060021)

BANDLA PUNITH (212219060037)

TABLE OF CONTENTS

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Humans are able to recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development. In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others.

In this project Gesture based Desktop automation system is developed.First the model is pretrained with the images of different hand gestures, such as a showing numbers with fingers as 0,1,2,3,4 . This project 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.Depending upon the prediction of gestures,certain browsing of images is done.For example,if the gesture prediction is 1,then the images is blurred;2, image is resized;3,image is rotated etc.

1.2 PURPOSE

It is used to browse the radiology images with help of hand gestures rather than using mouse,keyboard,etc thereby maintaining sterility during operations in hospitals.This prevents contamination and infection of the operating region as image is browsed with help of hand gesture.This also helps the doctors to complete the operation in short duration of time.

2. LITERATURE SURVEY

2.1 Existing problem

During operations in hospital,doctors find it difficult to perform operation by using mouse,keyboard,etc to browse the radiology images.It is necessary for the doctors to keep his hand clean during the entire operation,else it will lead to infection and contamination of the operating region of the patient.In order to overcome this,a gesture based tool for sterile browsing of radiology images is suggested.The existing gesture based tool does not predict all the gestures which are in different orientations and the distance between hand and the camera is small.

2. 2 References

1. **Gesture-controlled image system positioning for minimally invasive interventions**

Benjamin Fritsch, Thomas Hoffmann, André Mewes and Georg Rose

This work examines how a touchless interaction concept contributes to an efficient, direct, and sterile interaction workflow during CT-guided interventions. Two hand gesture sets were designed specifically under consideration of the clinical workflow and the hardware capabilities. These were used to change the position of an X-Ray tube and detector of a CT scanner without breaking sterility.

2. **A Gesture-based Tool for Sterile Browsing of Radiology Images**

Juan P. Wachs, Helman I. Stern, Yael Edan, Michael Gillam,Jon Handler,Craig Feied,Mark Smith

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.

3. **Intention, Context and Gesture Recognition for Sterile MRI Navigation in the Operating Room**

Mithun Jacob, Christopher Cange, Rebecca Packer, and Juan P. Wachs

Human-Computer Interaction (HCI) devices such as the keyboard and the mouse are among the most contaminated regions in an operating room (OR). This paper proposes a sterile, intuitive HCI to navigate MRI images using freehand gestures. The system incorporates contextual cues and intent of the user to strengthen the gesture recognition process.

4. **Vision Based Hand Gesture Recognition**

Pragati Garg, Naveen Aggarwal and Sanjeev Sofat

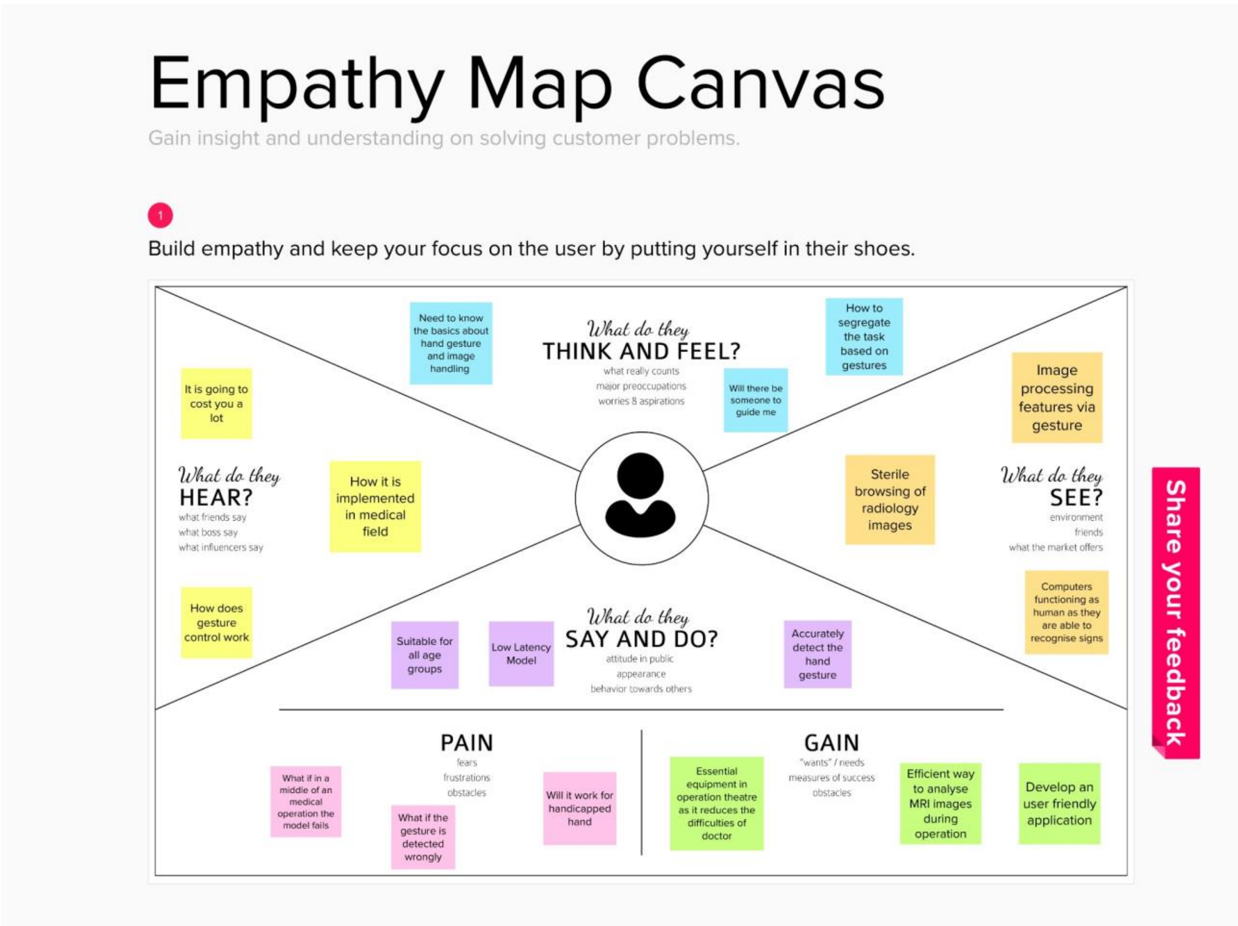
In this paper a review of vision based hand gesture recognition is presented. The existing approaches are categorized into 3D model based approaches and appearance based approaches, highlighting their advantages and shortcomings and identifying the open issues.

2.3 Problem Statement Definition

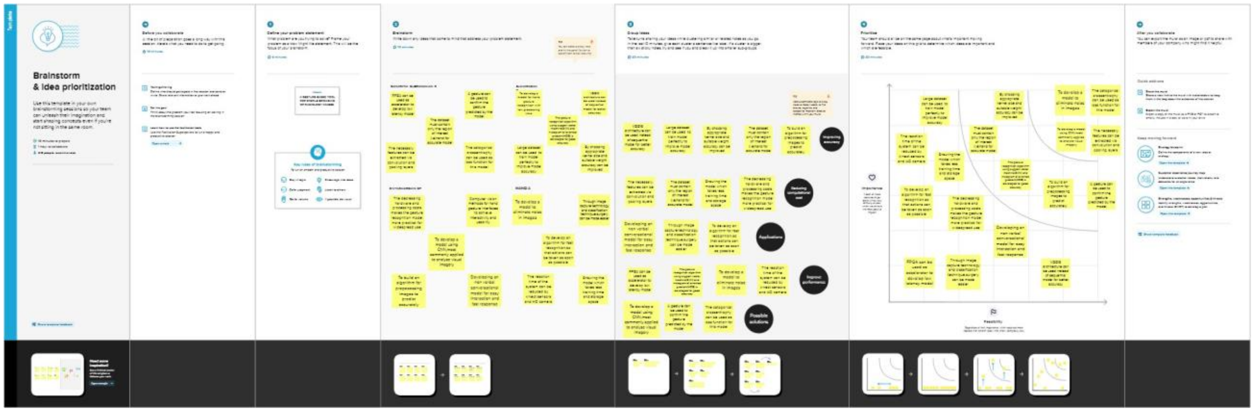
1. Huge data sets are needed to train the model, and it takes appropriate time and more storage space to store such data
2. Longer reaction time of the system leads to frustration and therefore to less concentration of the radiologist. However, most errors occurred when performing the confirmation gestures.Therefore an effective model is required.
3. Sometimes the participants held the hand palm rotated,this resulted in false prediction of gesture. In order to improve it,training dataset is increased with images of all orientation.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

1. Idea / Solution description

   Convolution Neural Network(CNN) architecture can be used to get an accurate model. The necessary features can be extracted by convolution and pooling layers without any human intervention.CNN gives high accuracy in image recognition problems.

2. Novelty / Uniqueness

   The novelty of our model lies in providing the best solution which is likely reporting the most accurate pose estimation. Even in situations with high obstruction,less clarity and pose estimation error, the tracking of the hand gesture and orientation by this model is accurate.

3. Social Impact / Customer Satisfaction

   The proposed model is cost efficient, user friendly and also enables the surgeon's focus of attention and high success rate in surgery which gives confident to the patient.

4. Business Model (Revenue Model)

   The decreasing hardware and processing costs makes the gesture recognition model more reliable. This makes the model more practical for widespread use in hospitals for easy interaction and fast response to make the surgery easier.

5. Scalability of the Solution

   With regard to image manipulation, contactless interfaces appear to be more suitable for interventional radiology units that already have a good level of scalability which increases accuracy.

3.4 Problem Solution fit

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

Project Design Phase-I - Problem Solution Fit

Team ID: PNT2022TMID35856

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

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

Doctors,nurses,surgeons and other medical professionals whose age is between 22 to 70.

6. CUSTOMER CONSTRAINTS CC

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

1)Uninterrupted power supply
2)High budget
3)Highly sophisticated cameras and processors.

5. AVAILABLE SOLUTIONS AS

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

1)Back up power supply can be used to overcome power cuts
2)Secondary camera can be used if the primary camera doesn't function well.

Explore AS, differentiate

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

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

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Human-Computer Interaction (HCI) devices such as the keyboard and the mouse are among the most contaminated regions in an operating room (OR). Therefore for sterile browsing of images of operating regions of a body,freehand gestures are used to adjust and browse the images

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.

Surgeons find it difficult to view and adjust the image of the operating area with their hand on the screen as their hands are not sterile during operation

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Expect on time notifications and remainders from the product manufacturer
Customers spend time on providing feedback.
Compare benefits over other service providers and explain their grievances to the product manufacturer regarding the same.

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

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

1)How to avoid contacting the most contaminated regions in operation theatre during operation and browse the images of the region of operation?

2)Seeing from neighbouring hospitals about how they effectively overcome the above problem during surgery

3)Reading about an efficient solution in news and coming up with an idea to implement it during surgery.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

BEFORE:

- **Fear about spreading of infections as there is a possibility of contamination**
- **Insecure**
- **Slow**

AFTER:

- **Satisfied**
- **Secure**
- **Fast and accurate**

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

For sterile browsing of images in operation theatre,a free hand gesture based deep learning model can be incorporated to browse the images with the help of a LCD monitor and a camera.In some cases the hand gesture shown by the surgeon may not be predicted by the deep learning model correctly.To overcome those situations, once the model recognises a hand gesture another hand gesture must be shown by the surgeon to confirm whether the predicted hand gesture is correct or not.This improves the overall accuracy of the model

8.CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

1)Customers upon facing difficulty should report it via Email or other platforms to the manufacturer.
2)Customers must fill the feedback form on regular basis.

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

1)Customers can improve the quality of camera for more accurate output.

2)Customers should place the monitor in an appropriate way such that the image is clearly visible to the surgeon and it is out of his/her reach.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR-1 User Registration

- Registration through Form
- Registration through Gmail
- Registration through LinkedIN

FR-2 User Confirmation

- Confirmation via Email
- Confirmation via OTP

FR-3 User Preferences

- Perform adjustment according to user's dominant hand.
- Accurate detection in all environment lightings.

FR-4 Object Location

- Upon detection, the system shall be able to compute the location of the object.

FR-5 Skin Colour Detection

- Filter out all object do not contain the colour of the skin.
- Focuses on hand detection and gesture recognition by eliminating background objects.

FR-6 User Satisfaction

- Satisfaction by High success rate.
- User friendly and ensure easy handling

4.2 Non-Functional requirements

NFR-1 Usability

- It is used by all technical and non-technical users as it is easily understandable with minimal instructions.

NFR-2 Security

- Access is permissible only in secure network, so there is less chance of insecurity.

NFR-3 Reliability

- This model shall be operable in all lighting conditions.
- Detect the user's hand at long distance.

NFR-4 Performance

- Reducing the memory requirements and computation complexity by using CNN gives better performance.

NFR-5 Availability

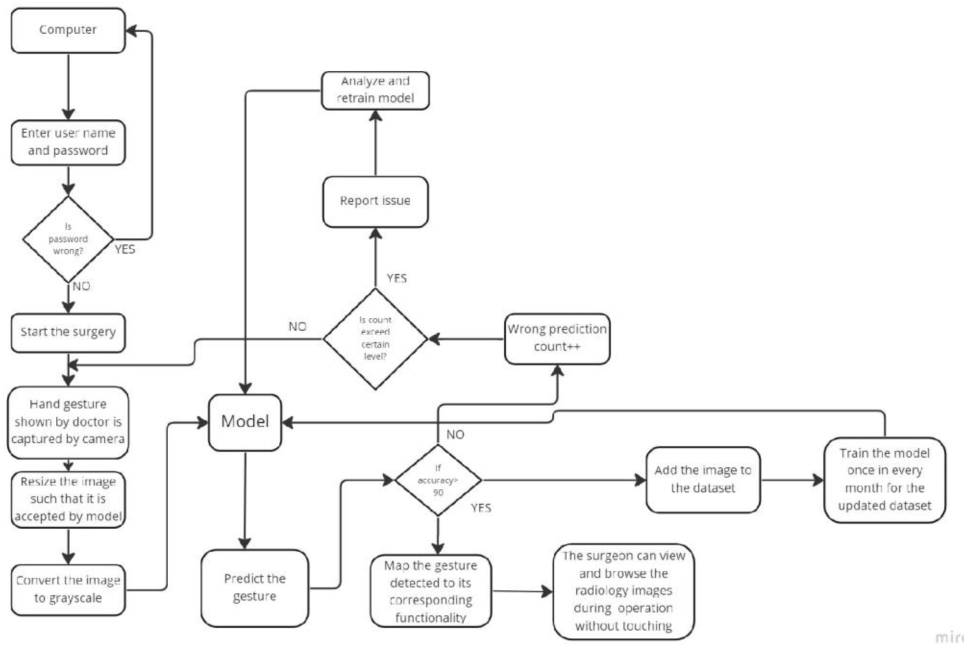
- Gesture recognition is an active research field, it is easily available.

NFR-6 Scalability

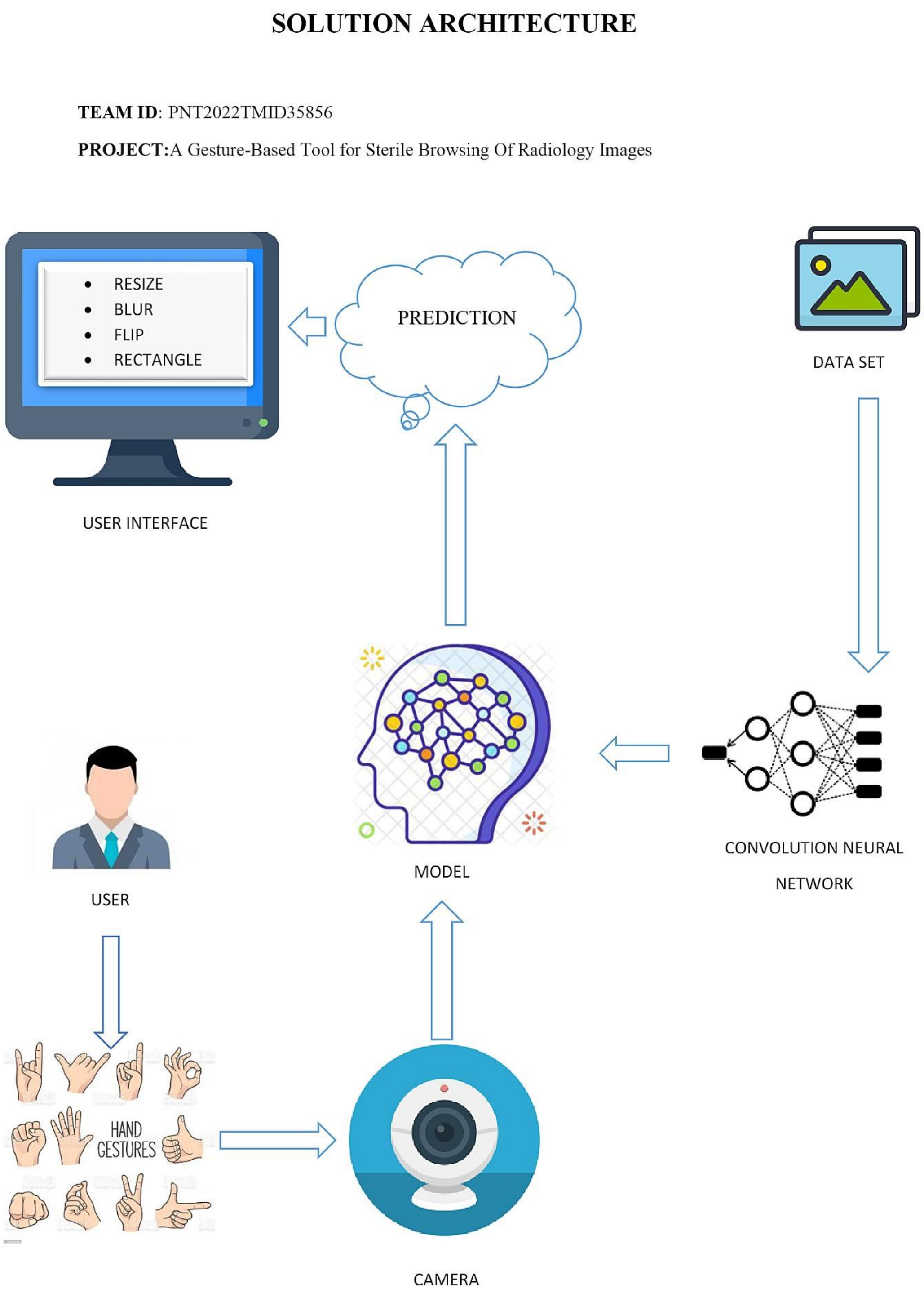
- Model is scaled by data augmentation and gesture recognition using CNN.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard	USN-3	As a user, I can see the various radiology image formatting methods out of which the necessary methods are selected.	I can register & access the dashboard.	High	Sprint-2
	Login	USN-4	As a user, I can access the application through Gmail.	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login and repeated usage	USN-5	As a user, I can log into the application by entering email & password	I can log in and view my dashboard	High	Sprint-1
	Web page details	USN-6	As a user, place the hand gesture in front of camera	I can see that the system has capture my hand gesture correctly	High	Sprint-2
Customer (Web user)	Upload the image in the web application	USN-7	As a user I must receive a correct hand gesture recognition as output	I can get the correct hand Gesture recognition.	High	Sprint -3
Customer Care Executive	Provide efficient customer support	USN-8	As a user, I need to get support from developers in case of queries and failure of service provided	I can have smooth user experiences and all the issues raised is sorted	Medium	Sprint -4
Administrator	Overview the entire process. Take all the responsibility and act bridge between users and developers	USN-9	We need to satisfy the customer needs in an efficient way and make sure any sort of errors are fixed	I can finish the work without any problems	High	Sprint - 4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	18 October 2022
Team ID	PNT2022TMID35856
Project Name	A Gesture-based Tool for Sterile Browsing of Radiology Images
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Building IBM Watson Assistant	USN-1	The entire source code is available in the IBM cloud	2	High	Ganapathi Subramanian R Divyadharshini S P
Sprint-1		USN-2	The dataset and model are available in cloud and should be trained in proper time gap	1	High	Elavarasan P Mansi A
Sprint-2	Modelling	USN-3	The model should be created such that it should be predict the hand gestures and do the appropriate function	2	Low	Ganapathi Subramanian R Divyadharshini S P
Sprint-2		USN-4	The processing time or response time should be reduced for fast response	2	Medium	Ganapathi Subramanian R Mansi A
Sprint-3	User Interface and Testing	USN-5	As a user ,I must have smooth gesture experience with good user interface satisfying all my expectation	1	High	Elavarasan P Divyadharshini S P
Sprint-3		USN-6	Testing the prototype created to ensure proper interpretation of messages and high level performance	2	High	Elavarasan P Mansi A
Sprint-4	Model Improvision		The dataset should be increased in order to improve accuracy	2	High	Ganapathi Subramanian R Divyadharshini S P
Sprint-4			The input image should have high resolution	1	Medium	Ganapathi Subramanian R Elavarasan P
			Gain information about the shortcomings from the feedback provided and improve service	1	Medium	Elavarasan P Mansi A

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

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

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

AV=sprint duration/velocity=20/6=3.33

6.2 Sprint Delivery Schedule

Sprint	Milestone
Sprint 1	1.Collection of required hand gesture images and building the dataset. 2.Uploading the dataset collected to the IBM Cloud.
Sprint 2	1. Developing a Model and verifying the accuracy. 2. Creating a secure database to store and access user information and images. 3. User will get the respective functionalities like zoom in or zoom out picture of operating part
Sprint 3	1. Users can access their dashboard and other details can be updated. 2. User show hand gesture in front of camera and the system itself can crop the gesture part alone in entire image
Sprint 4	1.Collection of user feedback and queries are done. 2.Constant updation of the portal and its respective backend work is done.

7. CODING & SOLUTIONING

Project Flow

1. User interacts with the UI (User Interface) to upload the image as input
2. Depending on the different gesture inputs different operations are applied to the input image.
3. Once model analyses the gesture, the prediction with operation applied on image is showcased on the UI.

To accomplish this, we have to complete all the activities and tasks listed below:

- Data Collection.
  - Collect the dataset or Create the dataset
- Data Pre processing
  - Import the ImageDataGenerator library

○ Configure ImageDataGenerator class

○ Apply ImageDataGenerator functionality to Trainset and Testset ●

Model Building

○ Import the model building Libraries

○ Initializing the model

○ Adding Input Layer

○ Adding Hidden Layer

○ Adding Output Layer

○ Configure the Learning Process

○ Training and testing the model

○ Save the Model

● Application Building

○ Create an HTML file

○ Build Python Code

Following software, concepts and packages are used in this project

● Anaconda navigator ●

Python packages:

○ open anaconda prompt as administrator

○ Type “pip install TensorFlow”

○ Type “pip install opencv-python”

○ Type “pip install flask”

CNN:

A convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery.

Opencv:

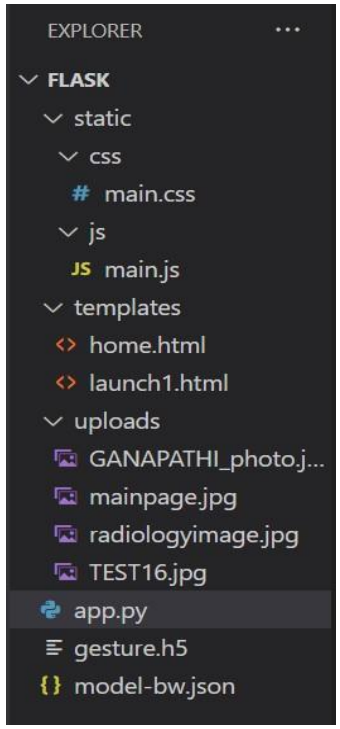
It is an Open Source Computer Vision Library which are mainly used for image processing, video capture and analysis including features like face detection and object detection.

Flask:

Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications

Project Structure

1. Dataset folder contains the training and testing images for training our model.
2. We are building a Flask Application which needs HTML pages stored in the templates folder and a python script app.py for server side scripting
3. We need the model which is deployed from IBM Cloud which is saved under the name gesture.h5
4. The static folder will contain js and cssfiles
5. Whenever we upload a image to predict, the upload images is saved in uploads folder.



Data Collection

ML depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.We have increased the dataset by changing its orientation such as rotation_range=30,horizontal_flip=True.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator,img_to_array,load_img
datagen = ImageDataGenerator(rotation_range=30,horizontal_flip=True,fill_mode='nearest')

for ix in range(1,100):
  img = load_img('/content/drive/MyDrive/handgesturedataset/dataset/train/5/'+str(ix)+'.jpg')
  x=img_to_array(img)
  x=x.reshape((1,)+x.shape)
  i=1;
  for batch in datagen.flow(x,save_to_dir='/content/drive/MyDrive/handgesturedataset/dataset/train/tr_5',save_prefix=str(ix),save_format='jpg'):
    i+=1;
    if i>4:
      break
```

Image Preprocessing

In this step we improve the image data that suppresses unwilling distortions or enhances some image features important for further processing, although perform some geometric transformations of images like rotation, scaling, translation etc

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255)#Normalisation
#Preprocessing the Training dataset
train_gen=train_datagen.flow_from_directory(
    'dataset/train',
    #Image size:128*128
    target_size=(128,128),
    batch_size=198,
    #Train Dataset has Grayscale images
    color_mode='grayscale',
    #Since output has classification class_mode='categorical'
    class_mode='categorical'
)
```

```
Found 2376 images belonging to 6 classes.
```

.

```python
#Test data preprocessing
from tensorflow.keras.preprocessing.image import ImageDataGenerator
#Normalisation
valid_datagen=ImageDataGenerator(rescale=1./255)
valid_gen=valid_datagen.flow_from_directory(
    'dataset/test',
    target_size=(128,128),
    batch_size=10,
    color_mode='grayscale',
    class_mode='categorical'
)
```

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

Model Building

In this step we build Convolutional Neural Networking which contains a input layer along with the convolution, maxpooling,hidden layers and finally a output layer.Dense layer is deeply connected neural network layer. It is most common and frequently used layer

```python
#Convolution Neural Network
#Importing tensorflow library
import tensorflow as tf
print(tf.__version__)
#Model Creation
model=tf.keras.Sequential([
                #Conv2D -To extract the essential features
                #Maxpooling - to compress the image without losing its features
                #Inupt shape of the image=128*128 pixels
                #relu - if x>0,return x;else return 0
                tf.keras.layers.Conv2D(16,(3,3),activation='relu',input_shape=(128,128,1)),
                tf.keras.layers.MaxPooling2D(2,2),
                tf.keras.layers.Conv2D(32,(3,3),activation='relu'),
                tf.keras.layers.MaxPooling2D(2,2),
                tf.keras.layers.Conv2D(16,(3,3),activation='relu'),
                tf.keras.layers.MaxPooling2D(2,2),
                #Takes the image and coverts it to a linear array-flatten
                tf.keras.layers.Flatten(),
                #Hidden layers
                #softmax-Activation function that predict multinomial probability
                tf.keras.layers.Dense(512,activation='relu'),#Hidden layer1:512 neurons
                tf.keras.layers.Dense(6,activation='softmax')#Output layer=No of classifications
```

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

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 126, 126, 16)      160

 max_pooling2d_6 (MaxPooling  (None, 63, 63, 16)       0
 2D)

 conv2d_7 (Conv2D)           (None, 61, 61, 32)        4640

 max_pooling2d_7 (MaxPooling  (None, 30, 30, 32)       0
 2D)

 conv2d_8 (Conv2D)           (None, 28, 28, 16)        4624

 max_pooling2d_8 (MaxPooling  (None, 14, 14, 16)       0
 2D)

 flatten_2 (Flatten)         (None, 3136)              0

 dense_4 (Dense)             (None, 512)               1606144

 dense_5 (Dense)             (None, 6)                 3078

=================================================================
Total params: 1,618,646
Trainable params: 1,618,646
Non-trainable params: 0
_________________________________________________________________
```

Configure The Learning Process

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

```
#Categorical crossentropy as output layer has 6 classes
#Optimiser-to reduce the weights or coefficients of hidden layer
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['Accuracy'])
```

Train The Model

1. Train the model with our image dataset.The 'fit_generator' functions used to train a deep learning neural network Arguments:
2. **steps_per_epoch** : it specifies the total number of steps taken from the generator as soon as one epoch is finished and next epoch has started. We can calculate the value of steps_per_epoch as the total number of samples in your dataset divided by the batch size.

3. **Epochs** : an integer and number of epochs we want to train our model for.
4. Validation_data can be either:
    1. an inputs and targets list
    2. a generator
    3. an inputs, targets, and sample_weights list

which can be used to evaluate the loss and metrics for any model after any epoch has ended.
5. **Validation_steps** :only if the validation_data is a generator then only this argument can be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.

```
[ ] #Train the model
    trainmodel=model.fit(
        train_gen,#Preprocessed Training dataset
        steps_per_epoch=12,#Total images in training dataset/batch_size of train dataset
        epochs=17,
        validation_data=valid_gen,#Preprocessed Test dataset
        validation_steps=3#Total images in test dataset/batch_size of test dataset
    )
```

```
[ ] Epoch 1/17
    12/12 [==============================] - 26s 2s/step - loss: 1.6598 - Accuracy: 0.2988 - val_loss: 1.2940 - val_Accuracy: 0.5333
    Epoch 2/17
    12/12 [==============================] - 25s 2s/step - loss: 1.0772 - Accuracy: 0.6120 - val_loss: 1.3332 - val_Accuracy: 0.4000
    Epoch 3/17
    12/12 [==============================] - 26s 2s/step - loss: 0.7170 - Accuracy: 0.7247 - val_loss: 0.6527 - val_Accuracy: 0.7667
    Epoch 4/17
    12/12 [==============================] - 25s 2s/step - loss: 0.5884 - Accuracy: 0.7715 - val_loss: 0.6410 - val_Accuracy: 0.7333
    Epoch 5/17
    12/12 [==============================] - 26s 2s/step - loss: 0.4581 - Accuracy: 0.8304 - val_loss: 0.6164 - val_Accuracy: 0.7000
    Epoch 6/17
    12/12 [==============================] - 26s 2s/step - loss: 0.3671 - Accuracy: 0.8754 - val_loss: 0.6048 - val_Accuracy: 0.7000
    Epoch 7/17
    12/12 [==============================] - 26s 2s/step - loss: 0.2795 - Accuracy: 0.9179 - val_loss: 0.4531 - val_Accuracy: 0.8000
    Epoch 8/17
    12/12 [==============================] - 26s 2s/step - loss: 0.2393 - Accuracy: 0.9242 - val_loss: 0.7577 - val_Accuracy: 0.6333
    Epoch 9/17
    12/12 [==============================] - 26s 2s/step - loss: 0.2202 - Accuracy: 0.9230 - val_loss: 0.4227 - val_Accuracy: 0.7333
    Epoch 10/17
    12/12 [==============================] - 26s 2s/step - loss: 0.1602 - Accuracy: 0.9554 - val_loss: 0.4835 - val_Accuracy: 0.7333
    Epoch 11/17
    12/12 [==============================] - 26s 2s/step - loss: 0.1400 - Accuracy: 0.9621 - val_loss: 0.3144 - val_Accuracy: 0.9000
    Epoch 12/17
    12/12 [==============================] - 26s 2s/step - loss: 0.0987 - Accuracy: 0.9785 - val_loss: 0.3224 - val_Accuracy: 0.9000
    Epoch 13/17
    12/12 [==============================] - 26s 2s/step - loss: 0.0628 - Accuracy: 0.9912 - val_loss: 0.3950 - val_Accuracy: 0.8667
    Epoch 14/17
    12/12 [==============================] - 26s 2s/step - loss: 0.0610 - Accuracy: 0.9899 - val_loss: 0.2909 - val_Accuracy: 0.9000
    Epoch 15/17
    12/12 [==============================] - 26s 2s/step - loss: 0.0459 - Accuracy: 0.9933 - val_loss: 0.2637 - val_Accuracy: 0.9667
    Epoch 16/17
    12/12 [==============================] - 26s 2s/step - loss: 0.0321 - Accuracy: 0.9975 - val_loss: 0.2892 - val_Accuracy: 0.9667
    Epoch 17/17
    12/12 [==============================] - 26s 2s/step - loss: 0.0250 - Accuracy: 0.9996 - val_loss: 0.3256 - val_Accuracy: 0.9333
```

Save the model

```
[ ] model.save('gesture.h5')
    model_json=model.to_json()
    with open("model-bw.json","w") as json_file:
      json_file.write(model_json)
```

Test The Model

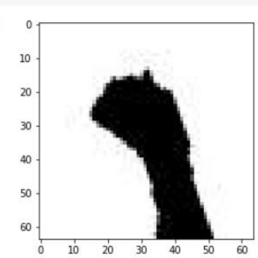
1. Evaluation is a process during development of the model to check whether the model is best fit for the given problem and corresponding data.
2. Load the saved model using load_model

```
[ ] from keras.models import load_model
    from keras.preprocessing import image
```

```
[ ] model=load_model('gesture.h5')
```

Plotting images:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
imgs=mpimg.imread(path)
imgplot=plt.imshow(imgs)
plt.show()
```



Taking an image as input and checking the results

```
[ ] img=image.load_img(path,color_mode='grayscale',target_size=(128,128))
    #image to array
    x=image.img_to_array(img)
```

```
[ ] x.shape
```

```
(128, 128, 1)
```

```
[ ] import numpy as np
    x=np.expand_dims(x,axis=0)
    x.shape
```

```
(1, 128, 128, 1)
```

```
[ ] pred=np.argmax(model.predict(x),axis=-1)
    #predicting the class
    index=['0','1','2','3','4','5']
```

```
pred
```

```
array([0])
```

```
[ ] result=str(index[pred[0]])
```

```
[ ] result
```

```
'0'
```

Application Building

After the model is trained in this particular step, we will be building our flask application which will be running in our local browser with a user interface.

Create HTML Pages

1. We use HTML to create the front end part of the web page.
2. Here, we created 2 html pages- home.html and launch.html
3. home.html displays home page.
4. launch.html accepts input from the user and predicts the values.

5. We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML pages.

```
∨ css
  # main.css
∨ js
  JS main.js
∨ templates
  <> home.html
  <> launch1.html
```

Build Python Code

1. Build flask file ‘app.py’ which is a web framework written in python for server-side scripting.
2. App starts running when “ name ” constructor is called in main.
3. render_template is used to return html file.
4. “GET” method is used to take input from the user.
5. “POST” method is used to display the output to the user.
6. Importing Libraries

```python
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
```

1. Creating our flask application and loading our model

```python
app = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model('gesture.h5')
print("Loaded model from disk")
```

2. Routing to the html Page

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

@app.route('/image1',methods=['GET','POST'])# routes to the index html
def image1():
    return render_template("launch1.html")
```

```python
@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def launch():
```

And the predict route is used for prediction and it contains all the codes which are used for predicting our results.

Firstly, inside launch function we are having the following things:

3. Getting our input and storing it
4. Grab the frames from the web cam.
5. Creating ROI
6. Predicting our results
7. Showcase the results with the help of opencv
8. Finally run the application

Getting our input and storing it

Once the predict route is called, we will check whether the method is POST or not if is POST then we will request the image files and with the help of os function we will be storing the image in the uploads folder in our local system.

```python
@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)
```

Grab the frames from the web cam

When we run the code a web cam will be opening to take the gesture input so we will be capturing the frames of the gesture for predicting our results.

```python
while True:
    _, frame = cap.read() #capturing the video frame values
    # Simulating mirror image
    frame = cv2.flip(frame, 1)
```

Creating ROI

A region of interest (ROI) is a portion of an image that you want to filter or operate on in some way. The toolbox supports a set of ROI objects that you can use to create ROIs of many shapes, such circles, ellipses, polygons, rectangles, and hand-drawn shapes. A common use of an ROI is to create a binary mask image.

```python
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)
```

Predicting our results

After placing the ROI and getting the frames from the web cam now its time to predict the gesture result using the model which we trained and stored it into a variable for the further operations.

```
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]}
```

Finally according to the result predicted with our model we will be performing certain operations like resize, blur , rotate etc.

```
key=cv2.waitKey(10)
if (key & 0xFF) == ord("s"):
    status = "Start"

if status == "Start":

    key=cv2.waitKey(10)

    if (key & 0xFF) == ord("d"):
        status = "Stop"

    if 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))
        rotated = cv2.resize(rotated, (300, 300))
        cv2.namedWindow("2 - Rotation", cv2.WINDOW_NORMAL)
        cv2.imshow("2 - Rotation", rotated)

    elif prediction[0][0]=='ONE':

        resized = cv2.resize(image1, (200, 200))
        cv2.namedWindow("1 - Fixed Resizing", cv2.WINDOW_NORMAL)
        cv2.imshow("1 - Fixed Resizing", resized)
```

```python
    elif prediction[0][0]=='THREE':
        blurred = cv2.GaussianBlur(image1, (21, 21), 0)
        blurred = cv2.resize(blurred, (200, 200))
        cv2.namedWindow("3 - Blurred", cv2.WINDOW_NORMAL)
        cv2.imshow("3 - Blurred", blurred)


    elif prediction[0][0]=='FOUR':

        resized = cv2.resize(image1, (400, 400))
        cv2.namedWindow("4 - Fixed Resizing", cv2.WINDOW_NORMAL)
        cv2.imshow("4 - Fixed Resizing", resized)

    elif prediction[0][0]=='ZERO':

        cv2.rectangle(image1, (150, 150), (250,250 ), (0, 0, 255), 2)
        image1 = cv2.resize(image1, (300, 300))
        cv2.namedWindow("0 - Rectangle", cv2.WINDOW_NORMAL)
        cv2.imshow("0 - Rectangle", image1)


    elif prediction[0][0]=='FIVE':

        gray = cv2.cvtColor(image1, cv2.COLOR_RGB2GRAY)
        cv2.namedWindow("5 - Gray Scale", cv2.WINDOW_NORMAL)
        gray = cv2.resize(gray, (300, 300))
        cv2.imshow("5 - Gray Scale", gray)
```

```python
    else:
        continue

key=cv2.waitKey(10)

val0=cv2.getWindowProperty("0 - Rectangle", cv2.WND_PROP_VISIBLE)
val1=cv2.getWindowProperty("1 - Fixed Resizing", cv2.WND_PROP_VISIBLE)
val2=cv2.getWindowProperty("2 - Rotation", cv2.WND_PROP_VISIBLE)
val3=cv2.getWindowProperty("3 - Blurred", cv2.WND_PROP_VISIBLE)
val4=cv2.getWindowProperty("4 - Fixed Resizing", cv2.WND_PROP_VISIBLE)
val5=cv2.getWindowProperty("5 - Gray Scale", cv2.WND_PROP_VISIBLE)


if (val4 and ((key & 0xFF) == ord("4"))):
    cv2.destroyWindow("4 - Fixed Resizing")
if (val1 and ((key & 0xFF) == ord("1"))):
    cv2.destroyWindow("1 - Fixed Resizing")
if (val0 and ((key & 0xFF) == ord("0"))):
    cv2.destroyWindow("0 - Rectangle")
if (val2 and ((key & 0xFF) == ord("2"))):
    cv2.destroyWindow("2 - Rotation")
if (val3 and ((key & 0xFF) == ord("3"))):
    cv2.destroyWindow("3 - Blurred")
if (val5 and ((key & 0xFF) == ord("5"))):
    cv2.destroyWindow("5 - Gray Scale")
if (key & 0xFF) == 27:
    cap.release()
    cv2.destroyAllWindows() # esc key
    break
```

Run The Application

At last, we will run our flask application

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

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

Run The app in local browser

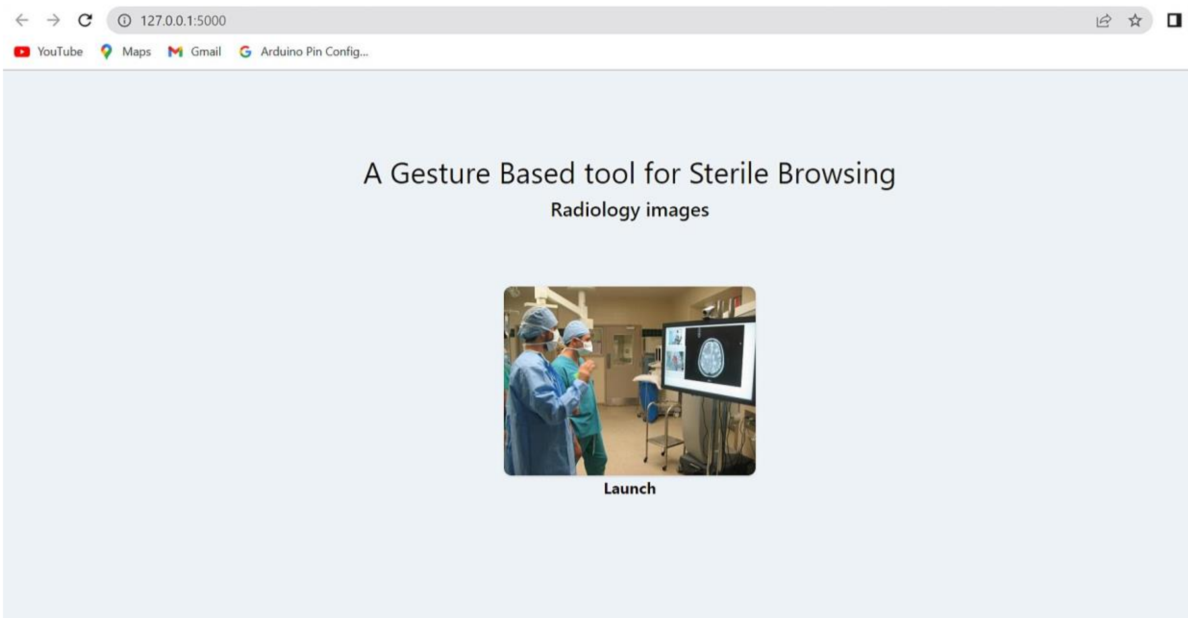
1. Open anaconda prompt from the start menu
2. Navigate to the folder where your python script is.
3. Now type “python app.py” command
4. Navigate to the localhost where you can view your web page

```
(base) D:\Flask>python app.py
2022-11-19 19:24:56.106321: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library
'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-11-19 19:24:56.107038: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not
have a GPU set up on your machine.
```

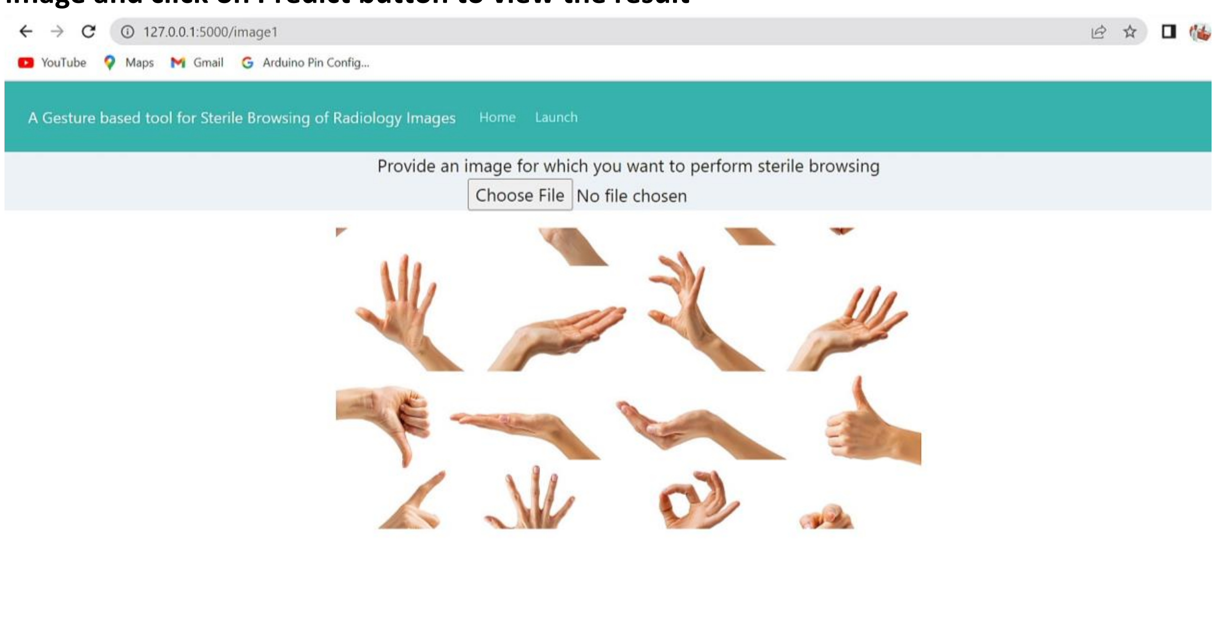
```
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Loaded model from disk
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

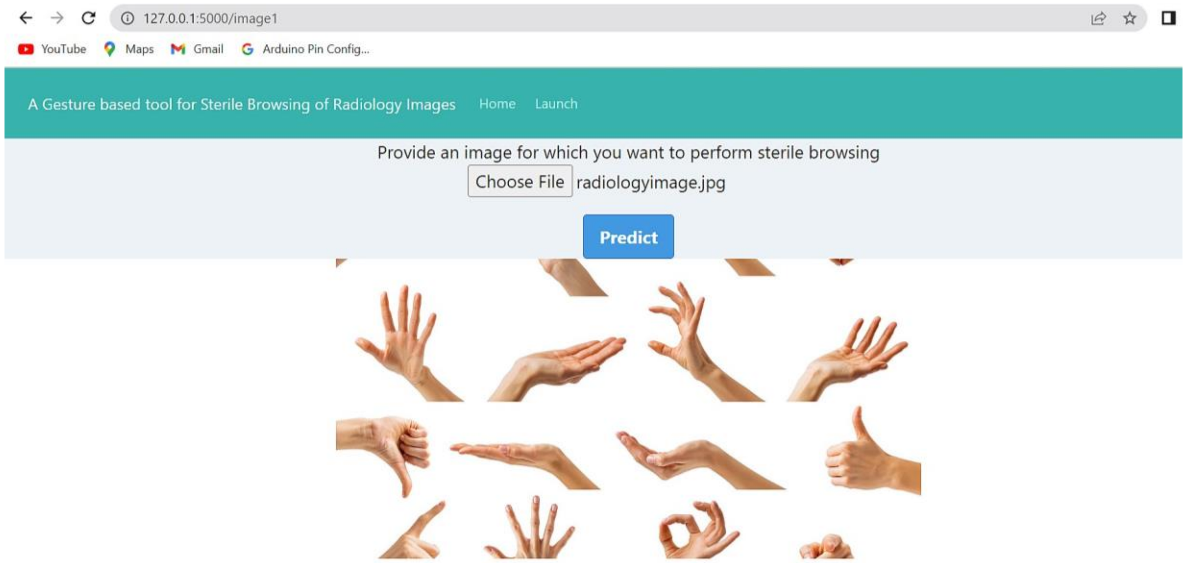
5. Navigate to the localhost (http://127.0.0.1:5000/)where you can view your web page.

Let’s see how our home.html page looks like:



Upload the image and click on Predict button to view the result





8. TESTING

8.1 User acceptance test

1.Purpose of Document

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

2.Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	3	5	7	23
Duplicate	2	0	0	0	2
External	5	4	0	1	10

Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	2	2	4
Won't Fix	0	3	0	0	3
Totals	26	12	12	28	80

3.Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	43	0	0	43
Security	2	0	0	2
Outsource Shipping	5	0	0	5
Exception Reporting	10	0	0	10
Final Report Output	2	0	0	2
Version Control	3	0	0	3

8.2 TestCase Report:

Date	03-Nov-22
Team ID	PNT2022TMID35856
Project Name	A Gesture-based Tool for Sterile E
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_OO 1	Functional	Home Page	Verify user is able to see the home page	HTML and JAVA SCRIPT	1.Enter URL and click go	http://localhost:5000/	home page should display.	Working as expected	Pass	1.Clear design 2.Easy to use	Yes	not applic able	Mansi A
HomePae _TC_OO2	UI	Home Page	Verify the UI elements "Home and Prediction"	HTML,CSS	1.Enter URL and click go 2.Click on Home, the page remains as it is 3.Click on launch ,it directs to launch page	http://localhost:5000/	Home and Launch icon works properly	Working as expected	Pass	1.User Friendly 2.Easy Navigation	Yes	not applic able	Divyadharshini S P
Launch_TC_OO3	Functional	Launch Page	Verify user is able to access predict page	HTML ,CSS	1.Enter URL and click go and it directs to home page 2.Click on "LAUNCH" 3.It direct to predict page	Not applicable	Launch page should display	Working as expected	Pass	1.Easy to use 2.Clear design	Yes	not applic able	Elavarasan P
Predict_TC_OO4	Functional	Predict page	Verify user is able to upload image and to predict the gestures correctly and gives the correct output	HTML,CSS and JAVA SCRIPT	1.Enter URL ,it will direct to home page 2.Click "LAUNCH" page 3.Upload a image by click on "Choose File" 4.Predict button will be enable 5.Click "Predict " button and application will start and produce the output	Not applicable	if predict is pressed,the application will start and correct output will be predicted	Working as expected	Pass	1.Clear and Easy access of the model	Yes	not applic able	Ganapathi Subramanian R
Model_TC_OO5	Functional	Model	Upload image and using webcam,gesture input is given,and correct gesture is predicted	Python	1.In "predict" upload image and click in "predict" and click on predict webcam wii be opened	Not applicable	Based on the given gesture output is predicted and image is predicted and image is processed by results	Working as expected	pass	Good accuracy	yes	not applic able	Elavarasan P

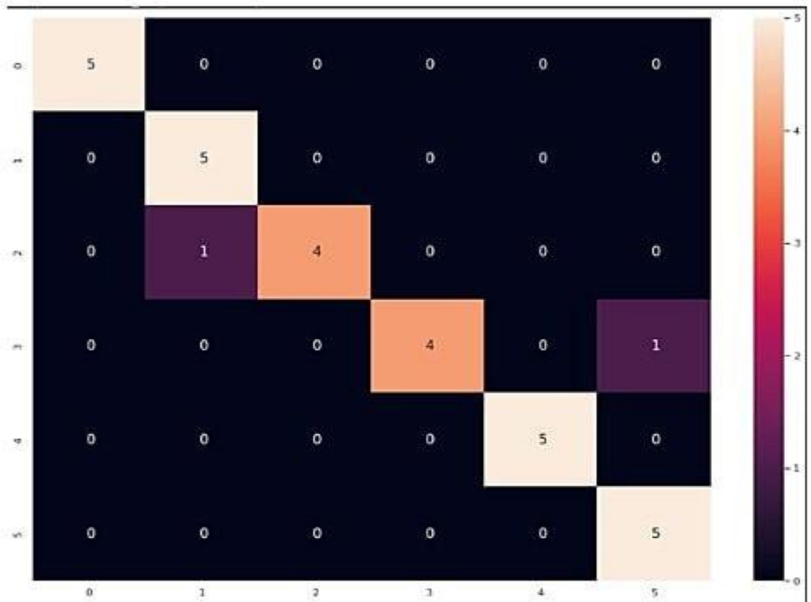
9. RESULTS

9.1 Performance Metrics

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	Model: "sequential" Layer (type) Output Shape Param # conv2d (Conv2D) (None, 126, 126, 16) 160 max_pooling2d (MaxPooling2D) (None, 63, 63, 16) 0 conv2d_1 (Conv2D) (None, 61, 61, 32) 4640 max_pooling2d_1 (MaxPooling 2D) (None, 30, 30, 32) 0 conv2d_2 (Conv2D) (None, 28, 28, 16) 4624 max_pooling2d_2 (MaxPooling 2D) (None, 14, 14, 16) 0 flatten (Flatten) (None, 3136) 0 dense (Dense) (None, 512) 1606144 dense_1 (Dense) (None, 6) 3078 Total params: 1,618,646 Trainable params: 1,618,646 Non-trainable params: 0
2.	Accuracy	Training Accuracy - 99.96 Validation Accuracy - 93.33	loss: 0.0250 - Accuracy: 0.9996 - val_loss: 0.3256 - val_Accuracy: 0.9333

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix : Accuray Score-:93.3% Classification Report :	CONFUSION MATRIX  ACCURACY SCORE: `Accuracy: 93.33333333333333 %` CLASSIFICATION REPORT:

<table>
<tr><td></td><td></td><td></td><td>CLASSIFICATION REPORT:
<pre>
Gesture Recognition model accuracy:0.9333
              precision   recall f1-score   support

           0   1.00   1.00   1.00   5
           1   0.83   1.00   0.91   5
           2   1.00   0.80   0.89   5
           3   1.00   0.80   0.89   5
           4   1.00   1.00   1.00   5
           5   0.83   1.00   0.91   5

    accuracy   0.93   30
   macro avg   0.94   0.93   0.93   30
weighted avg   0.94   0.93   0.93   30
</pre></td></tr>
<tr><td>2.</td><td>Tune the Model</td><td>Hyperparameter Tuning

Validation Method</td><td>
<pre>
[ ] import tensorflow as tf
    print(tf.__version__)
    model=tf.keras.Sequential([
                tf.keras.layers.Conv2D(16,(3,3),activation='relu',input_shape=(128,128,1)),
                tf.keras.layers.MaxPooling2D(2,2),
                tf.keras.layers.Conv2D(32,(3,3),activation='relu'),
                tf.keras.layers.MaxPooling2D(2,2),
                tf.keras.layers.Conv2D(16,(3,3),activation='relu'),
                tf.keras.layers.MaxPooling2D(2,2),
                tf.keras.layers.Flatten(),
                tf.keras.layers.Dense(512,activation='relu'),
                tf.keras.layers.Dense(6,activation='softmax')
    ])
</pre>
<pre>
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['Accuracy'])
</pre></td></tr>
</table>

10. ADVANTAGES & DISADVANTAGES

Advantages:

Major advantage of this tool is that it helps to maintain the sterility of the environment. It is also easy to use and is quicker than the existing methods to browse images. It can also be performed even if the surgeon is a bit far away from the system, this helps to save time. The tool does not need the person using it to have an apparatus or any devices on them to use it. They can simply move their hands to browse through the images.

Disadvantages:

The tool can be quite expensive as it requires cameras and other expensive devices to capture images and process it.

11. CONCLUSION

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

12. FUTURE SCOPE

- The tool can be made quicker by increasing the recognition speed.
- More number of gestures can be added thereby increasing this tool's functionality and useability for different purposes.
- Tracking of both hands can be added to increase the set of commands.
- Voice commands can also be added to further increase the functionality.

13. APPENDIX

Source Code:

https://github.com/IBM-EPBL/IBM-Project-27746-1660064699/tree/main/Project%20Development%20Phase

GitHub & Project Demo Link

Github: https://github.com/IBM-EPBL/IBM-Project-27746-1660064699