A GESTURE BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

PROJECT REPORT

Submitted by

TEAM ID: PNT2022TMID29001

TEAM LEADER: T.G.GOKHUL (411719205018)

TEAM MEMBER: TA.M.AKSHAYA (411719205006)

TEAM MEMBER: K.SRIRAM (411719205042)

TEAM MEMBER: G.SWETHA (411719205049)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

PRINCE SHRI VENKATESHWARA PADMAVATHY ENGINEERING COLLEGE, PONMAR

ANNA UNIVERSITY: CHENNAI-600 025

NOVEMBER-2022

TABLE OF CONTENT

CHAPTER-1

INTRODUCTION

1.1 PROJECT OVERVIEW

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

In this project Gesture based Desktop automation, First the model is trained pre-trained on the images of different hand gestures, such as showing numbers with fingers as 1,2,3,4. This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the gesture predicts is 0 - then images is converted into rectangle, 1 - image is Resized into (200,200), 2 - image is rotated by -45°, 3 - image is blurred, 4 - image is Resized into (400,400), 5 - image is converted into grayscale etc.

1.2 PURPOSE

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.

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.

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

CHAPTER-2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Humans have the ability to recognize body and sign language but computers don't have this ability. Humans can recognize sign language because of the combination of vision and synaptic interactions with the brain. To make computers recognize sign language we need to replicate this skill to computers. Through the position and shape of the center of the palm and the fingers we can obtain certain information. The gesture can be both static and dynamic. Static hand gestures are obtained by analyzing the shape of the hand. Dynamic hand gestures are obtained by analyzing hand movements. The ability to spontaneously identify gestures without delay in hand motion is the problem. Through real-time hand gesture detection, we overcome these problems. Processing speed, image processing techniques and different recognition algorithms are used in this real-time hand gesture detection. In this project, the model is first pre-trained on the images of different hand gestures, such as showing numbers with fingers as 1, 2, 3, 4. This model uses the integrated webcam to capture the video frame. The image captured in the video frame is compared with the pre-trained model and gesture is identified.

2.2 REFERENCES

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

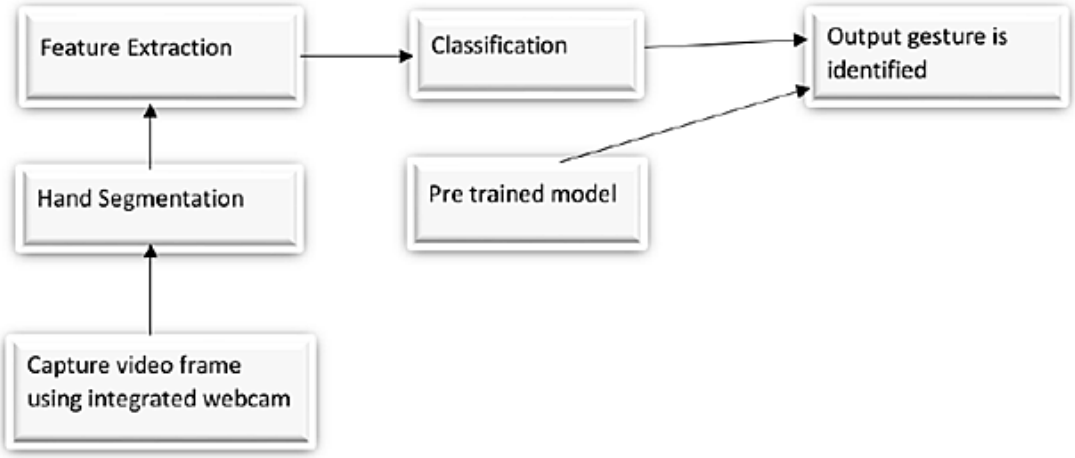
8. Lewis JR. Psychometric evaluation of an after scenario questionnaire for computer usability studies: The ASQ *SIGCHI Bulletin* 1991;23:78-81. [Google Scholar]

2.3 PROBLEM STATEMENT DEFINITION

To develop a CNN based classifier model, which would be trained on our training data. We train a CNN based model to recognize the hand gesture. The training data include images that capture the hand gestures of 1,2,3,4,5 and 0. The image is resized without much loss of information and used for training a CNN based model. We use Python Flask to provide an interactive platform for our model. This project would help the doctors in operation theatres where physical contact between persons should be avoided in order to be sterilised and also

prevent from any infections.

Humans have the ability to recognize body and sign language but computers don’t have this ability. Humans can recognize sign language because of the combination of vision and synaptic interactions with the brain. To make computers recognize sign language we need to replicate this skill to computers.

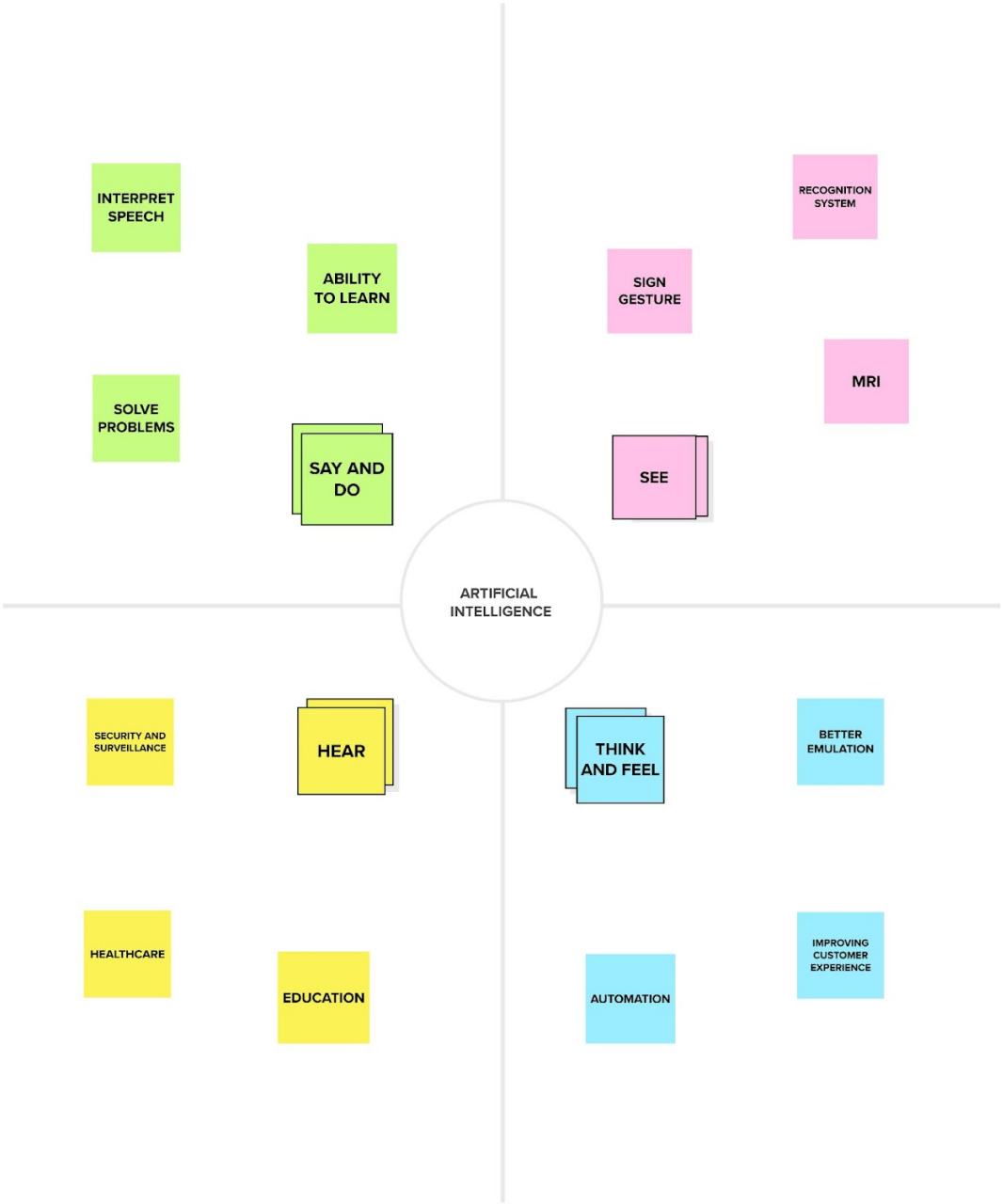


Through the position and shape of the centre of the palm and the fingers we can obtain certain information. In this project, the model is first pre-trained on the images of different hand gestures, such as showing numbers with fingers as 1, 2, 3, 4. This model uses the integrated webcam to capture the video frame. The image captured in the video frame is compared with the pre-trained model and gesture is identified.

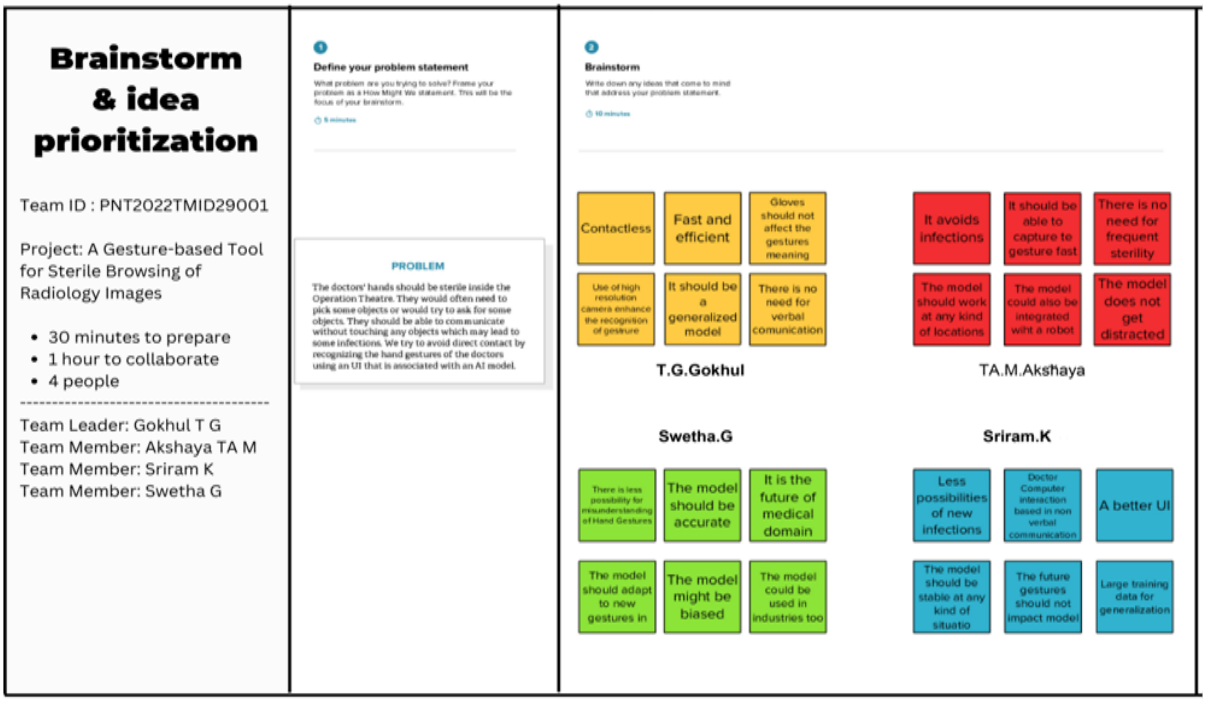
CHAPTER-3

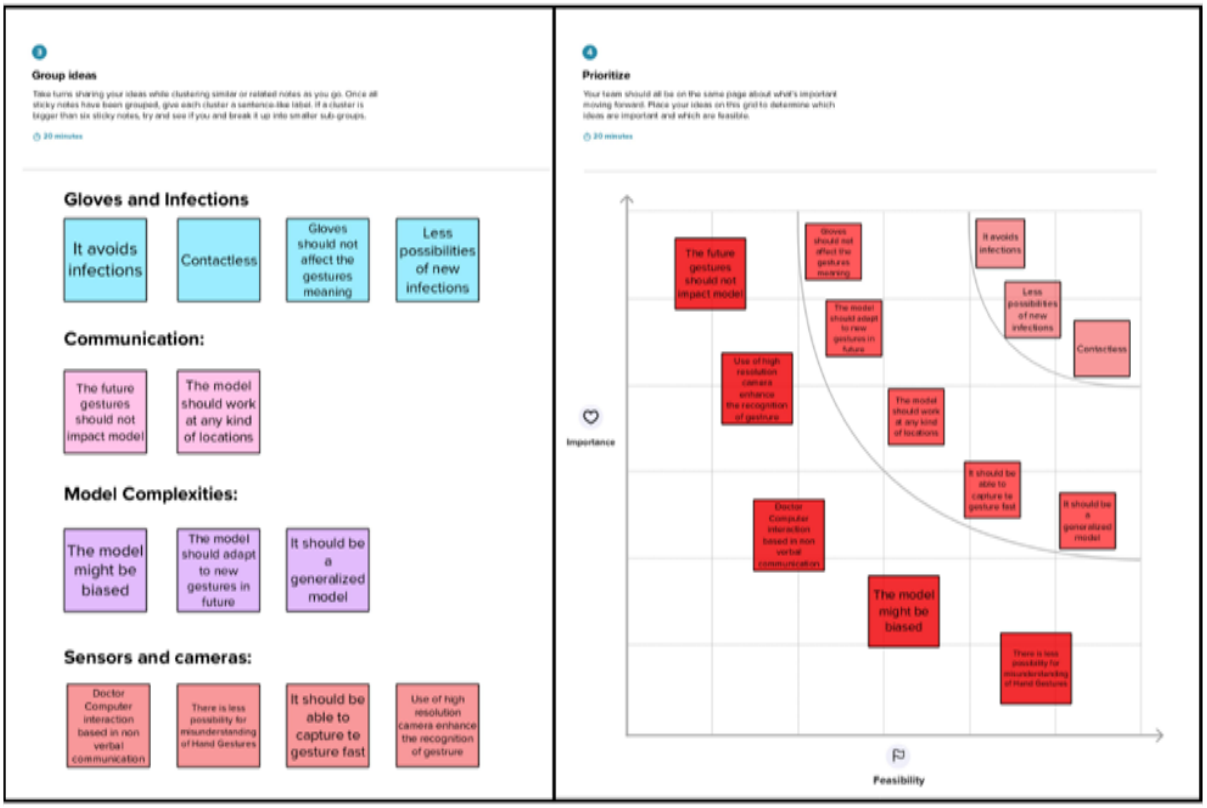
IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING

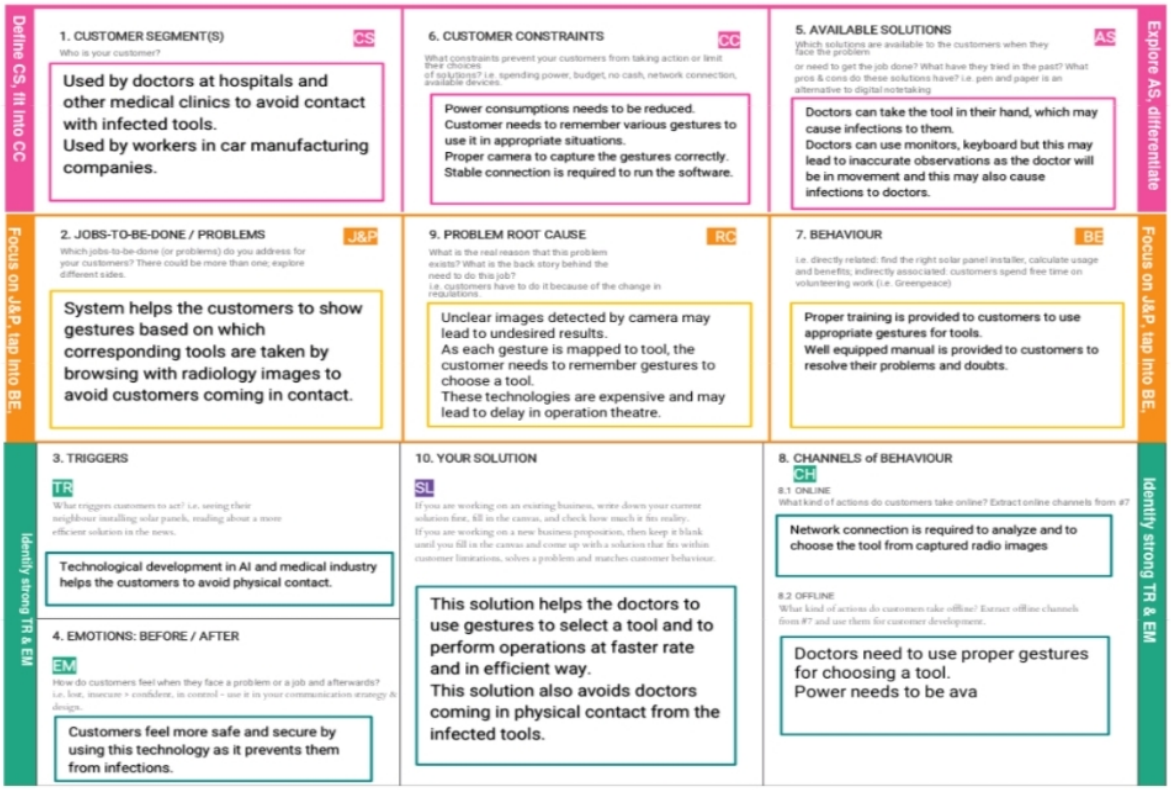




3.3 PROPOSED SOLUTION

SI NO	PARAMETER	DESCRIPTION
1	Problem Statement	To design an ML model to identify and classify the hand gestures.
2	Idea / Solution description	To develop a CNN based classifier model, which would be trained on our training data.
3	To develop a CNN based classifier model which would be trained on our training data.	We train a CNN based model to recognize the hand gesture. The training data include images that capture the hand gestures of 1,2,3,4,5 and 0. The image is resized without much loss of information and used for training a CNN based model. We use Python Flask to provide an interactive platform for our model.
4	Social Impact / Customer Satisfaction	This project would help the doctors in operation theaters where physical contact between persons should be avoided in order to be sterilized and also prevent from any infections.
5	Business Model (Revenue Model)	It can be sold as an open-source service to all the hospitals as a non-profitable work.
6	Scalability of the Solution	The model could also be extended to other real world classifying problems like cancer detection from X-ray, COVID detection using X-ray images, mask detection, face detection etc.

3.4 PROBLEM SOLUTION FIT



CHAPTER-4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Identifying User Gestures	The user gestures are identified using the images of gestures captured by the camera
FR-2	Deployment in Cloud	The trained Deep Learning Model is deployed in cloud, which could be accessed anywhere around the world
FR-3	User Interface	The user interface, which helps in the Human Computer Interaction is designed
FR-4	Gestures related to the Application Domain	The model should be trained with the gestures related to the application domain.

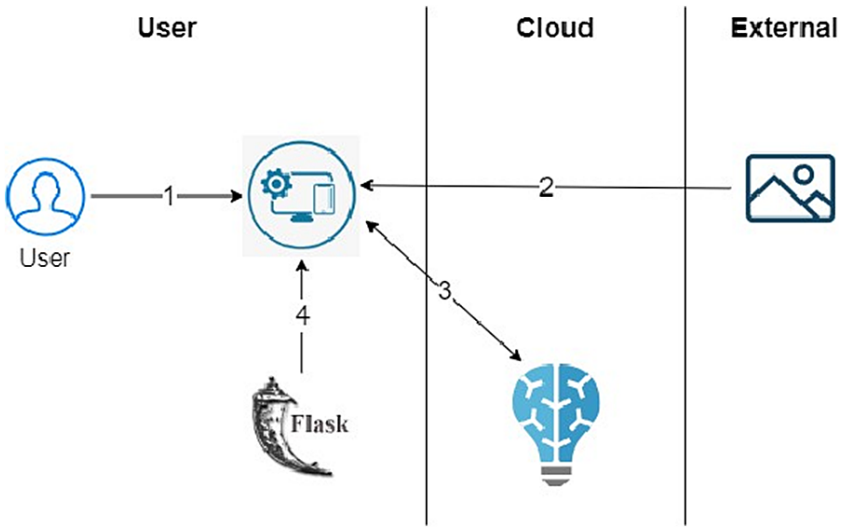
4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user interface which acts as an intermediate between the user and the DL Model which is deployed in the cloud
NFR-2	**Security**	The model deployed in the cloud should be accessible only by the approved users and it should be inaccessible by the attackers or the terrorists
NFR-3	**Reliability**	The tool or the system is 95% reliability for a year
NFR-4	**Performance**	The tool or the system should respond with the accurate response within 4-5 seconds
NFR-5	**Availability**	The model deployed in the cloud must be available to 99.8% of the people over a month during working hours
NFR-6	**Scalability**	The model deployed in the cloud must be accessible by over 10,00,000 people trying to access it using the user interface

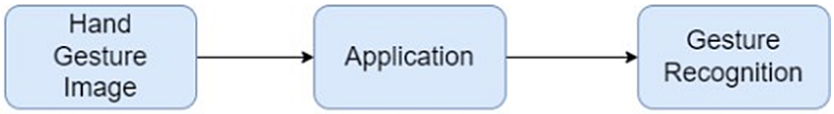
CHAPTER - 5

PROJECT DESIGN

5.1.DATA FLOW DIAGRAMS

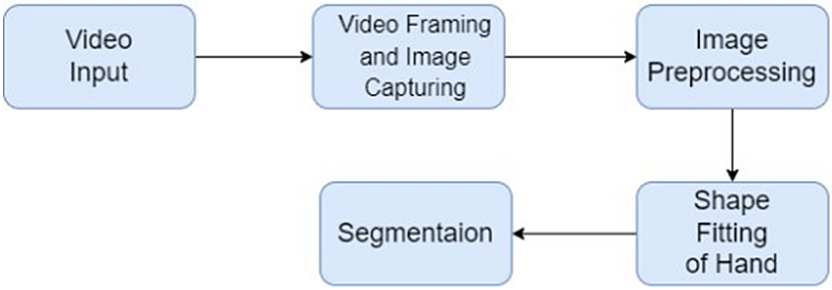


Data Flow Diagram - Level 0



Data Flow Diagram - Level 1

Data Flow Diagram - Level 2

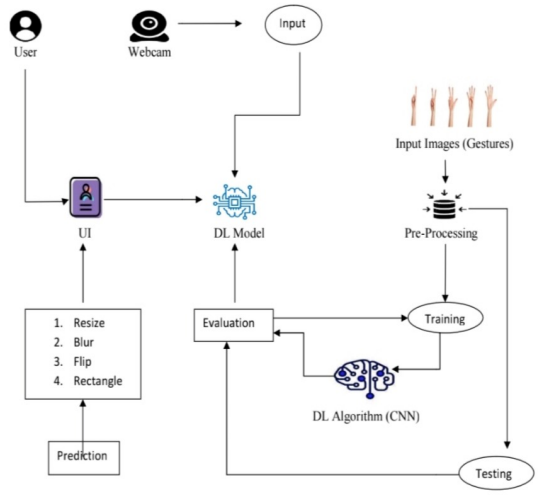


5.2 SOLUTION & TECHNICAL ARCHITECTURE

In this project Gesture based Desktop automation ,First the model is trained pre-trained on the images of different hand gestures, such as showing numbers with fingers as 1 ,2,3,4 . This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the gesture predicted is 1 then images are blurred;2, image is resized;3,image is rotated etc.Gesture operations are initiated by a calibration mode in which a skin color model of the user's hand or glove, under local lighting, is constructed.

In a browse mode, superimposed over the image of the camera's scene is a rectangular frame called the “neutral area.” Movements of the hand across its boundary constitute directional browser commands. When a doctor/surgeon wishes to browse the image database, the hand is moved rapidly out of the “neutral area” toward any of four directions, and then back again.System helps the customer to show gestures based on which corresponding tools are taken by browsing with radiology images to avoid customers coming in contact.

TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Launch Web App deployed in cloud	USN-1	As a user, I can launch the webapp where I can upload the images for recognition	I can upload the images for classification	High	Sprint-4
	Prediction	USN-2	As a user, I can get the predicted results from the model deployed in the cloud	I can resize the radiology image, blur the image, flip based on the hand gesture	High	Sprint-4
	Deployment of webapp in the cloud	USN-3	As a user, I need the webapp to be accessible all over the world	I can access the webapp deployed in the IBM cloud	Medium	Sprint-3
	Deployment of AI model in the cloud	USN-4	As a user, I need the AI model to be accessible all over the world	I can access the model deployed in the IBM cloud	Medium	Sprint-3
	Model Building	USN-5	As a user, I need an AI model which could classify or recognize the hand gestures	I can get the prediction from the AI model	Medium	Sprint-1
	User Interface Building	USN-6	As a user, I need a web app for human computer interaction	I get User Interface for interaction with the model	Medium	Sprint-2

CHAPTER-6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

SI. NO	MILESTONE	ACTIVITIES	DATE
1	**Preparation Phase**	Pre-requisites	22 - 27 Aug 2022
		Prior knowledge	
		Project Structure	
		Project Flow	
		Project Objectives	
		Registrations	
		Environment Set-up	
2	**Ideation Phase**	Literature Survey	29 Aug - 3rd Sept 2022
		Empathy Map	5 - 10th Sept 2022
		Ideation	12 - 17 Sept 2022
3	**Project Design Phase -I**	Proposed Solution	19 - 24 Sept 2022
		Problem Solution Fit	26 Sept - 01 Oct 2022
		Solution Architecture	
4	**Project Design Phase -II**	Customer Journey	3 - 8 Oct 2022
		Requirement Analysis	10 - 15 Oct 2022
		Data Flow Diagrams	
		Technology Architecture	
5	**Project Planning Phase**	Milestones & Tasks	17 - 22 Oct 2022
		Sprint Schedules	
6	**Project Development Phase**	Sprint-1	24 - 29 Oct 2022
		Sprint-2	31 Oct - 5 Nov 2022
		Sprint-3	7 - 12 Nov 2022
		Sprint-4	14 - 19 Nov 2022

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint - 1	Launching Software	USN - 1	As a user, I can launch the developed software.	1	Low	Akshaya TA M Gokhul T G Sriram K Swetha G
Sprint - 1	Access UI	USN - 2	As a user, I can use the software and operate on the UI.	1	Medium	Akshaya TA M Gokhul T G Sriram K Swetha G
Sprint - 2	Launching Camera	USN - 3	As a user, I can open the camera from the software to perform gestures.	1	Low	Akshaya TA M Gokhul T G Sriram K Swetha G
Sprint - 2	Upload images from local system	USN - 4	As a user, I can upload images to the software from the local system.	2	Low	Akshaya TA M Gokhul T G Sriram K Swetha G
Sprint - 3	Perform gestures	USN - 5	As a user, I can perform various gestures with respect to system specification for processing.	2	Medium	Akshaya TA M Gokhul T G Sriram K Swetha G
Sprint - 4	Display output	USN - 6	As a user, I can see the sterile browsed image with respect to the gestures performed, displayed on the screen.	2	High	Akshaya TA M Gokhul T G Sriram K Swetha G

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

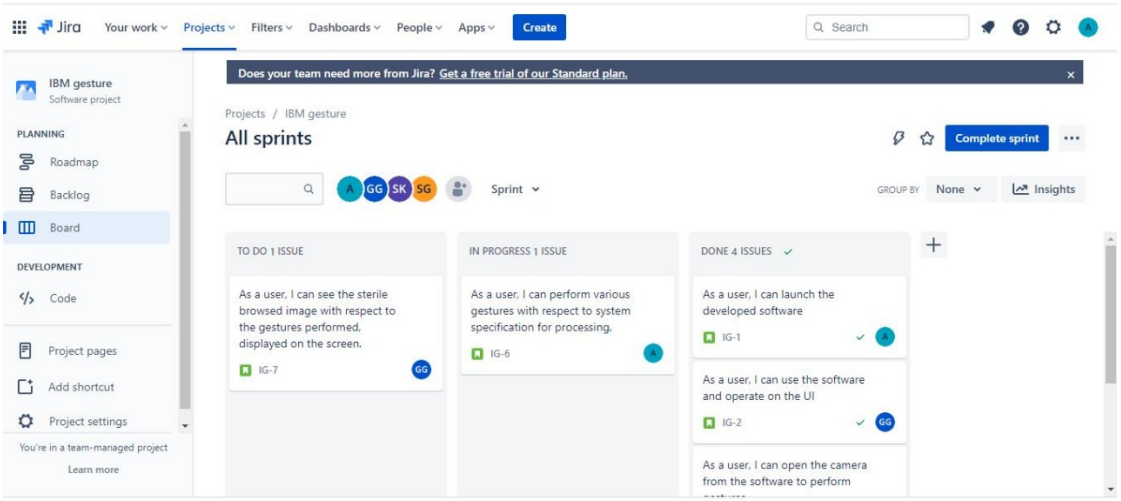
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

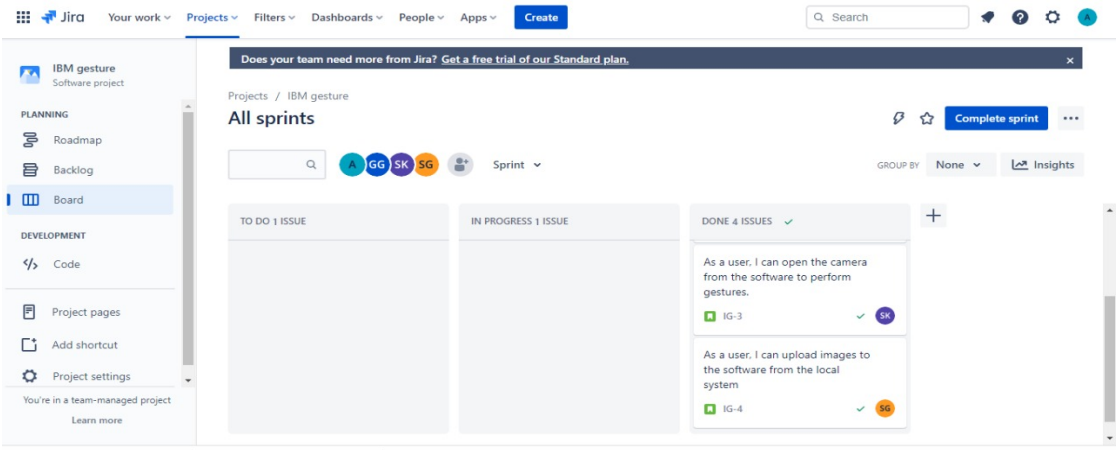
Velocity:

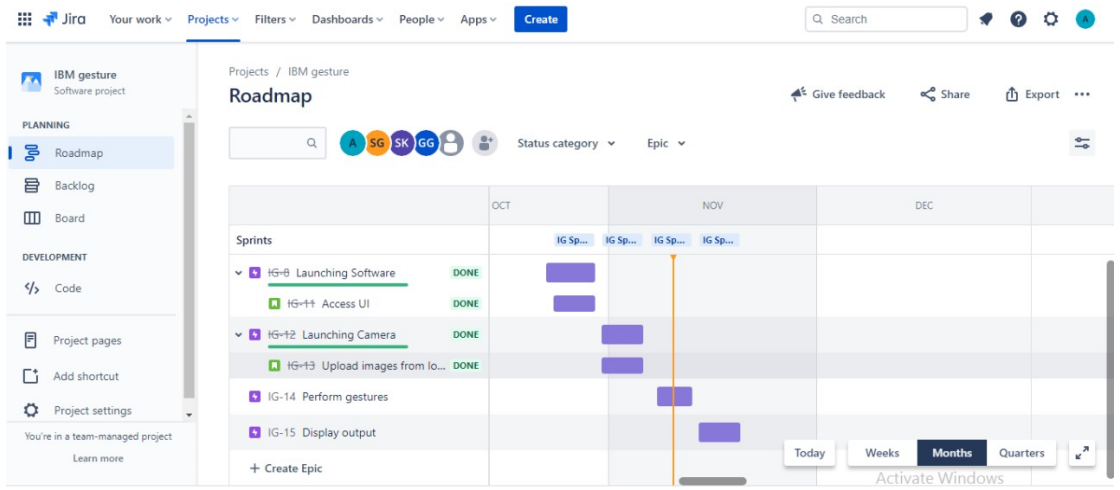
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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

6.3 REPORTS FROM JIRA







CHAPTER-7

CODING & SOLUTION

7.1 PROJECT STRUCTURE

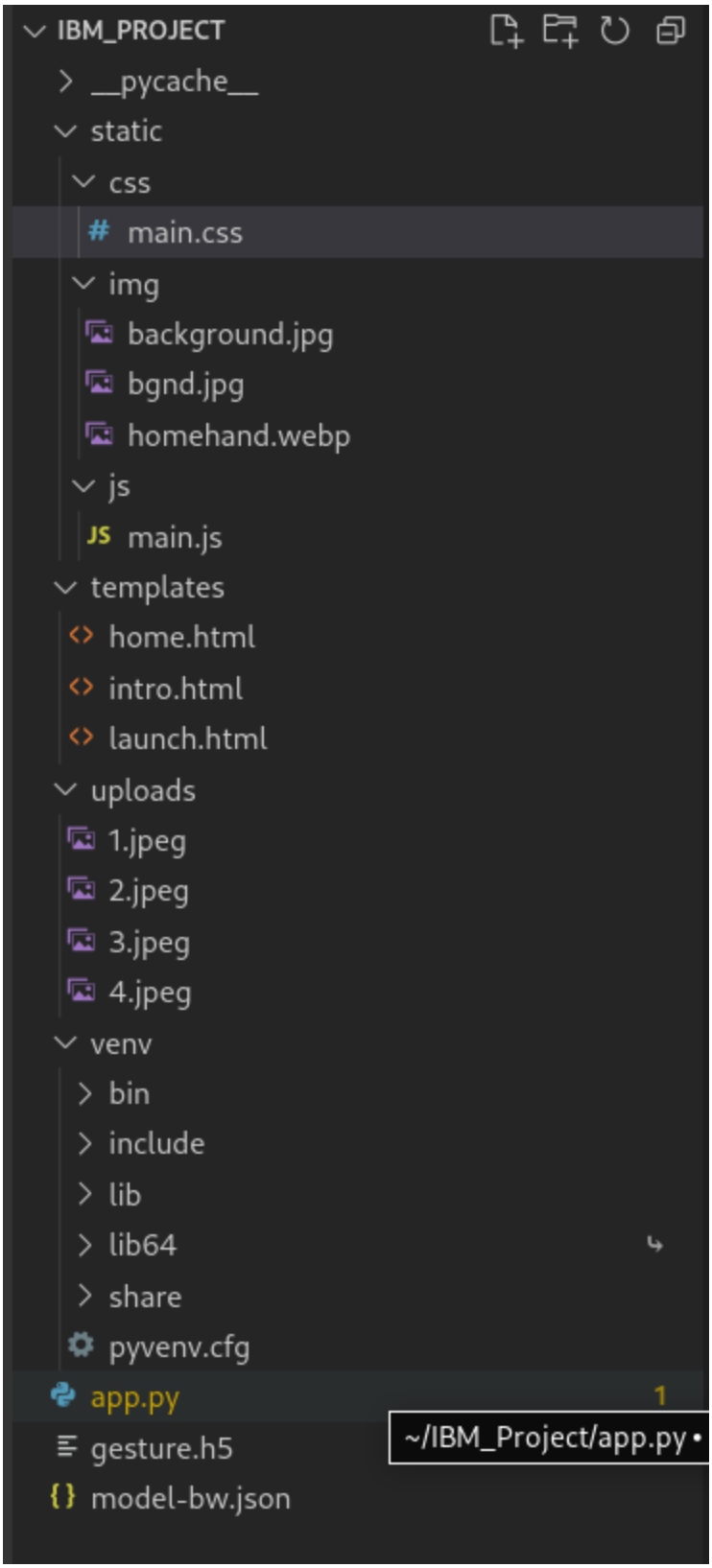
- The Dataset folder contains the training and testing images for training our model.

- 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.

- We need the model which is saved and the saved model in this content is gesture.h5.

- The static folder will contain js and css files.

- Whenever we upload an image to predict, that image is saved in the uploads folder.


IBM_PROJECT
__pycache__
static
css
main.css
img
background.jpg
bgnd.jpg
homehand.webp
js
main.js
templates
home.html
intro.html
launch.html
uploads
1.jpeg
2.jpeg
3.jpeg
4.jpeg
venv
bin
include
lib
lib64
share
pyvenv.cfg
app.py
gesture.h5
model-bw.json
~/IBM_Project/app.py

7.2 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.

Apply ImageDataGenerator

```
train_data= train_datagen.flow_from_directory(r'D:\Dataset\train',

target_size=(80,80),batch_size=8,class_mode='categorical',subset='training',color_
mode='grayscale')
test_data = test_datagen.flow_from_directory(r'D:\Dataset\test',

target_size=(80,80),batch_size=8,class_mode='categorical',color_mode='grayscale
')
```

Configure ImageDataGenerator

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
```

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,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
train_data= train_datagen.flow_from_directory(r'D:\Dataset\train',

target_size=(80,80),batch_size=8,class_mode='categorical',subset='training',color_mode='grayscale')
test_data = test_datagen.flow_from_directory(r'D:\Dataset\test',

target_size=(80,80),batch_size=8,class_mode='categorical',color_mode='grayscale')
```

Import the ImageDataGenerator Library

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

7.3 MODEL BUILDING

In this step we build Convolutional Neural Networking which contains a input layer along with the convolution, max pooling and finally a output layer.

Adding CNN Layers

```
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())
```

Adding Dense Layers

Dense layer is deeply connected neural network layer. It is most common and frequently used layer.

```
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'))
```

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.

Configure The Learning Process

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.

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

Optimization is an important process which optimize the input weights by comparing the prediction and the loss function. Here we are using Adam optimizer

Metrics is used to evaluate the performance of your model. It is similar to loss function, but not used in training process

7.4 TRAIN THE MODEL

Train the model with our image dataset.

fit_generator functions used to train a deep learning neural network

```
checkpoint = ModelCheckpoint(r'D:\IBM Project\gesture.h5',
                             monitor='val_loss',save_best_only=True,verbose=3)

earlystop = EarlyStopping(monitor = 'val_loss', patience=7, verbose= 3,
restore_best_weights=True)
learning_rate = ReduceLROnPlateau(monitor= 'val_loss', patience=7, verbose= 3, )

callbacks=[checkpoint,earlystop,learning_rate]

model.fit_generator(train_gen,
                    epochs=25,
                    steps_per_epoch=18000//32,
                    validation_data=test_gen,
                    callbacks=callbacks,
```

verbose = 1,validation_steps=3600//32)

Arguments:

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.

Epochs : an integer and number of epochs we want to train our model for.

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.

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.

Test the model

Evaluation is a process during development of the model to check whether the model is best fit for the given problem and corresponding data.

Load the saved model using load_model

```
from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

model=load_model(r'D:\IBM Project\gesture.h5')

categories=['0','1','2','3','4','5']

test_img=image.load_img(r'Downloads\OIP4
image.jpg',target_size=(80,80),color_mode='grayscale')

pixels=image.img_to_array(test_img)

pixels=np.expand_dims(pixels,axis=0)

prediction=model.predict(pixels)

print(np.argmax(pred))
```

Plotting images:

Taking an image as input and checking the results. By using the model we are predicting the output for the given input image. The predicted class index name will be printed here.

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

- We use HTML to create the front end part of the web page.
- Here, we created 3 html pages- home.html, intro.html and index6.html
- Home.html displays the home page.
- Intro.html displays introduction about the hand gesture recognition
- Index.html accepts input from the user and predicts the values.
- We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML pages.

```
<body>
<p style="background-image: url('/static/img/bgnd.jpg');">
<h1 style="color: rgb(255, 255, 255);">
  <table style="width:100%">
    <tr>
      <th><img src="/static/img/background.jpg" alt="NOT AVAILABLE" width="300" height="300"></th>
      <th> <iframe width="780" height="440" src="https://www.youtube.com/embed/nD621G8u6oc?start=3&loop=1&autoplay=1&mute=1&controls=0">
      </iframe><br></th>
      <th></th>
    </tr>
  </table>
    <br>
    <span>HAND   GESTURE   RECOGNITION</span>
  </br>
    <span>OF</span>
  </br>
    <span>RADIOLOGY   IMAGES</span>
  </br>
    <span>THROUGH</span>
  </br>
    <span>STERILE   BROWSING</span>

</h1>

<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:#ffffff; padding-top:1%;padding-left:5%;"><b>HAND GESTURE RECOGNITION</b></div>
  <div class="topnav-right"style="padding-top:0.5%;color:white">
    <a  class="active" href="{{ url_for('home')}}"><u>Home</u></a>
    <a  class="active" href="{{ url_for('intro')}}">Introduction</a>
    <a  class="active" href="{{ url_for('image1')}}">Launch</a>
  </div>
</div>
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:#ffffff; padding-top:1%;padding-left:5%;"><b>Project done for IBM - Team ID</b></div>
</body>
```

Build Python Code

- Build flask file ‘app.py’ which is a web framework written in python for server-side scripting.
- App starts running when the “ name ” constructor is called in main.
- render_template is used to return an 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

```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

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


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


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

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


@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web
def launch():
    if request.method == 'POST':
        print("inside image")
        f = request.files['image']

        basepath = os.path.dirname(__file__)
        file_path = os.path.join(basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        print(file_path)
        cap = cv2.VideoCapture(0)
        while True:
            _, frame = cap.read() #capturing the video frame values
            # Simulating mirror image
```

Creating our flask application and loading our model

Routing to the html Page

The above three routes are used to render the home, introduction and the index html pages. 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:

- Getting our input and storing it
- Grab the frames from the webcam.
- Creating ROI
- Predicting our results
- Showcase the results with the help of opencv
- Finally run the application

Getting our input and storing it

Once the predicted route is called, we will check whether the method is POST or not. If it 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.

Grab the frames from the webcam

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

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 as circles, ellipses, polygons, rectangles, and hand-drawn shapes. A common use of an ROI is to create a binary mask image.

Predicting our results

After placing the ROI and getting the frames from the webcam now it's time to predict the gesture result using the model which we trained and stored it into a variable for the further operations.
Finally according to the result predicted with our model we will be performing certain operations like resize, blur , rotate etc.

Run The Application

At last, we will run our flask application
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. Then it will run on localhost:5000

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

CHAPTER-8

TESTING

8.1 TEST CASES

				Date	9-Nov-22				
				Team ID	PNT2022TMID29001				
				Project Name	Project - A Gesture Based Tool For				
				Maximum Marks	4 marks				
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	https://gesture.com/	Login/Signup popup should display	Working as expected	Pass
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	https://gesture.com/	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: aksvm@gmail.com password: bsnl123	User should navigate to user account homepage	Working as expected	Pass
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: aksvm@gmail password: bsnl123	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: aksvm@gmail.com password: bsnl12367868678678876	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: aksvm password: bsnl167865	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass

8.2 USER ACCEPTANCE TESTING

Acceptance Testing
UAT Execution & Report Submission

Date	09 November 2022
Team ID	PNT2022TMID29001
Project Name	Project - A Gesture Based Tool For Steril Browsing of Radiology Images
Maximum Marks	4 Marks

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	12	4	4	5	25
Duplicate	1	2	2	0	5
External	4	3	1	0	8
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	28	16	15	27	86

3. Test Case Analysis

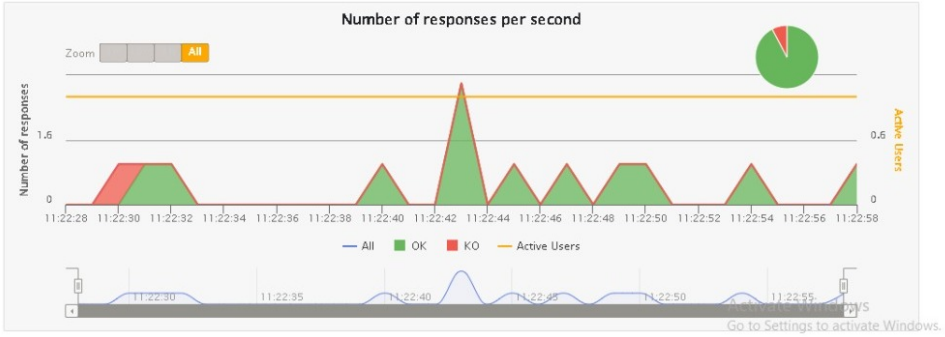
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

CHAPTER-9
RESULTS
9.1 PERFORMANCE METRICS

Expand all groups | Collapse all groups

Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
All Requests	13	12	1	8%	0.419	5	2458	9384	15885	18415	19048	4995	6058
request_0	1	0	1	100%	0.032	2458	2458	2458	2458	2458	2458	2458	0
request_5	1	1	0	0%	0.032	9	9	9	9	9	9	9	0
request_6	1	1	0	0%	0.032	8	8	8	8	8	8	8	0
request_7	1	1	0	0%	0.032	5	5	5	5	5	5	5	0
bootstrap.min.css	1	1	0	0%	0.032	9881	9881	9881	9881	9881	9881	9881	0
popper.min.js	1	1	0	0%	0.032	19048	19048	19048	19048	19048	19048	19048	0
jquery.min.js	2	2	0	0%	0.065	6784	8084	8734	9254	9358	9384	8084	1300
bootstrap.min.js	1	1	0	0%	0.032	13777	13777	13777	13777	13777	13777	13777	0
icon?family=Material+Icons	1	1	0	0%	0.032	3325	3325	3325	3325	3325	3325	3325	0
main.css	1	1	0	0%	0.032	113	113	113	113	113	113	113	0
main.js	1	1	0	0%	0.032	134	134	134	134	134	134	134	0
request_8	1	1	0	0%	0.032	10	10	10	10	10	10	10	0



CHAPTER - 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Know fundamental concepts and techniques of Convolutional Neural Network (CNN).
- Gain a broad understanding of image data
- Know how to pre-process/clean the data using different data pre-processing techniques.
- Know how to build a web application using the Flask framework.

DISADVANTAGES

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

CHAPTER - 11
CONCLUSION

- In this project , we have used a Convolutional Neural Network to first train the model on the images of different hand gestures , like showing numbers with fingers like 0,1,2,3,4, etc.
- After uploading the image , our portal uses the integrated webcam to capture the video using OpenCV.
- The gesture captured in the videoframe is compared with the pretrained model and the gesture is identified.

CHAPTER - 12
FUTURE SCOPE

- We can also enhance this project such that it can translate the hand gestures into powerpoint.
- Removal of wrist band and considering another reference point and also black background.
- A hand gesture system for MRI manipulation in an EMR image database called “Gestix” was tested during a brain biopsy surgery. This system is a real-time hand-tracking recognition technique based on color and motion fusion.
- In an in vivo experiment, this type of interface prevented a surgeon's focus shift and change of location while achieving rapid intuitive interaction with an EMR image database.
- The results of two usability tests (contextual and individual

interviews) and a satisfaction questionnaire indicated that the “Gestix” system provided a versatile method that can be used in the OR to manipulate medical images in real-time and in a sterile manner.

- We are now considering the addition of a body posture recognition system to increase the functionality of the system, as well as visual tracking of both hands to provide a richer set of gesture commands.
- For example, pinching the corners of a virtual image with both hands and stretching the arms would represent an image zoom-in action.
- In addition, we wish to assess whether a stereo camera will increase the gesture recognition accuracy of the system.
- A more exhaustive comparative experiment between our system and other human–machine interfaces, such as voice, is also left for future work.

CHAPTER - 13
APPENDIX

13.1 SOURCE CODE

```
import mediapipe as mp
import numpy as np
from flask import Flask, render_template, request
import cv2
import os
from keras.models import load_model
from werkzeug.utils import secure_filename


app = Flask(__name__)
@app.route("/")
def home():
    return render_template("home.html")

@app.route("/process", methods=['GET', 'POST'])
def process():
    if request.method == 'POST':
        upload_image = request.files['upload_image']
        basepath=os.path.dirname(__file__)

file_path=os.path.join(basepath,'static',secure_filename(upload_image.filename))
        upload_image.save(file_path)
        model1 = load_model('gesture.h5')
        mpHands = mp.solutions.hands
```

```
    hands = mpHands.Hands(max_num_hands=1, min_detection_confidence=0.5,
min_tracking_confidence=0.5)
    mpDraw = mp.solutions.drawing_utils
    cap = cv2.VideoCapture(0)
    while True:
        _, frame = cap.read()

        h, w, c = frame.shape

        frame = cv2.flip(frame, 1)
        framergb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        result = hands.process(framergb)
        res = ''

        if result.multi_hand_landmarks:
            landmarks = []
            for handslms in result.multi_hand_landmarks:
                x_max = 0
                y_max = 0
                x_min = w
                y_min = h
                for lm in handslms.landmark:
                    x = int(lm.x * w)
                    y = int(lm.y * h)

                    landmarks.append([x, y])
                    if x > x_max:
                        x_max = x
```

```
                if x < x_min:
                    x_min = x
                if y > y_max:
                    y_max = y
                if y < y_min:
                    y_min = y
            cv2.rectangle(frame, (x_min - 5, y_min - 5), (x_max + 5, y_max + 5),
(0, 255, 0), 2)
            framegray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            hand = framegray[y_min - 5:y_max + 5, x_min - 5:x_max + 5]
            hand = cv2.resize(hand, (128, 128))
            hand = hand / 255
            hand = hand.reshape(128, 128, 1)
            hand = np.expand_dims(hand, axis=0)
            mpDraw.draw_landmarks(frame, handslms,
mpHands.HAND_CONNECTIONS)
            prediction = model1.predict(hand)
            res = np.argmax(prediction)
            image1 = cv2.imread(file_path)
            image1= cv2.resize(image1,(300,300))
            if res==1:
                resized = cv2.resize(image1, (200, 200))
                cv2.imshow("Resizing", resized)
                key=cv2.waitKey(3000)

                if (key & 0xFF) == ord("1"):
                    cv2.destroyWindow("Resizing")
```

```
elif res==2:
    blurred = cv2.GaussianBlur(image1, (21, 21), 0)
    cv2.imshow("Blurred", blurred)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("3"):
        cv2.destroyWindow("Blurred")

elif res==3:
    (h, w, d) = image1.shape
    center = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(center, -45, 1.0)
    rotated = cv2.warpAffine(image1, M, (w, h))
    cv2.imshow("OpenCV Rotation", rotated)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("2"):
        cv2.destroyWindow("OpenCV Rotation")


elif res==4:
    cv2.rectangle(image1, (480, 170), (650, 420), (0, 0, 255), 2)
    cv2.imshow("Rectangle", image1)
    cv2.waitKey(0)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("0"):
        cv2.destroyWindow("Rectangle")


else:
    continue
```

```
            cv2.imshow("Output", frame)

            if cv2.waitKey(1) == ord('q'):
                break

        cap.release()
        cv2.destroyAllWindows()

    return render_template("index.html")


@app.route("/intro")
def intro_page():
    return render_template("intro.html")

@app.route("/index")
def index_page():
    return render_template("index.html")
```

13.2 GITHUB LINK AND DEMO LINK

Github

https://github.com/IBM-EPBL/IBM-Project-11633-1659336530.git

Demo link

https://www.youtube.com/watch?v=QV12ytil6P8