A GESTURE BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

TEAM ID: PNT2022TMID19926
BATCH NUMBER: BATCH-2[B2M4E]

PROJECT DONE BY:

-727819TUIT122	RAGULBALAJI S
-727819TUIT133	SHAMYUKTHA K
-727819TUIT137	SNEHA T
-727819TUIT140	SUJAN KARTHI N
-727819TUIT155	YOGHESH S

1. INTRODUCTION

1.1 Project Overview

In this project we use gestures to browse images obtained during radiology. Gestures refer to nonverbal form of communication made using hands. A major challenge involved in this process is to provide doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work. Keyboards and pointing 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. Humans can recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development.

In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others. In this project Gesture based Desktop automation, First the model is trained by pre-trained on the images of different hand gestures, such as a showing numbers with fingers as 1,2,3,4.This model uses the integrated webcam to capture the video frame. The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the gesture predicts is 0 - then images is converted into rectangle, 1 - image is Resized 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 Gray scale etc.

1.2 Purpose

It is used to browse through the images obtained using radiology using hand gestures rather than using a mouse,keyboard,etc, thereby maintaining sterility.

2.LITERATURE SURVEY

2.1 Existing Problem

In the existing system, doctors need to move from their place while operating patients. And also they cannot be able to complete their work on-time. In order to view the images for every time they need to move to the monitor and thereby make contact with the hardware peripheral devices.So it leads to the wide spread of diseases.

2.2 References

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2410001/

https://pubmed.ncbi.nlm.nih.gov/18451034/

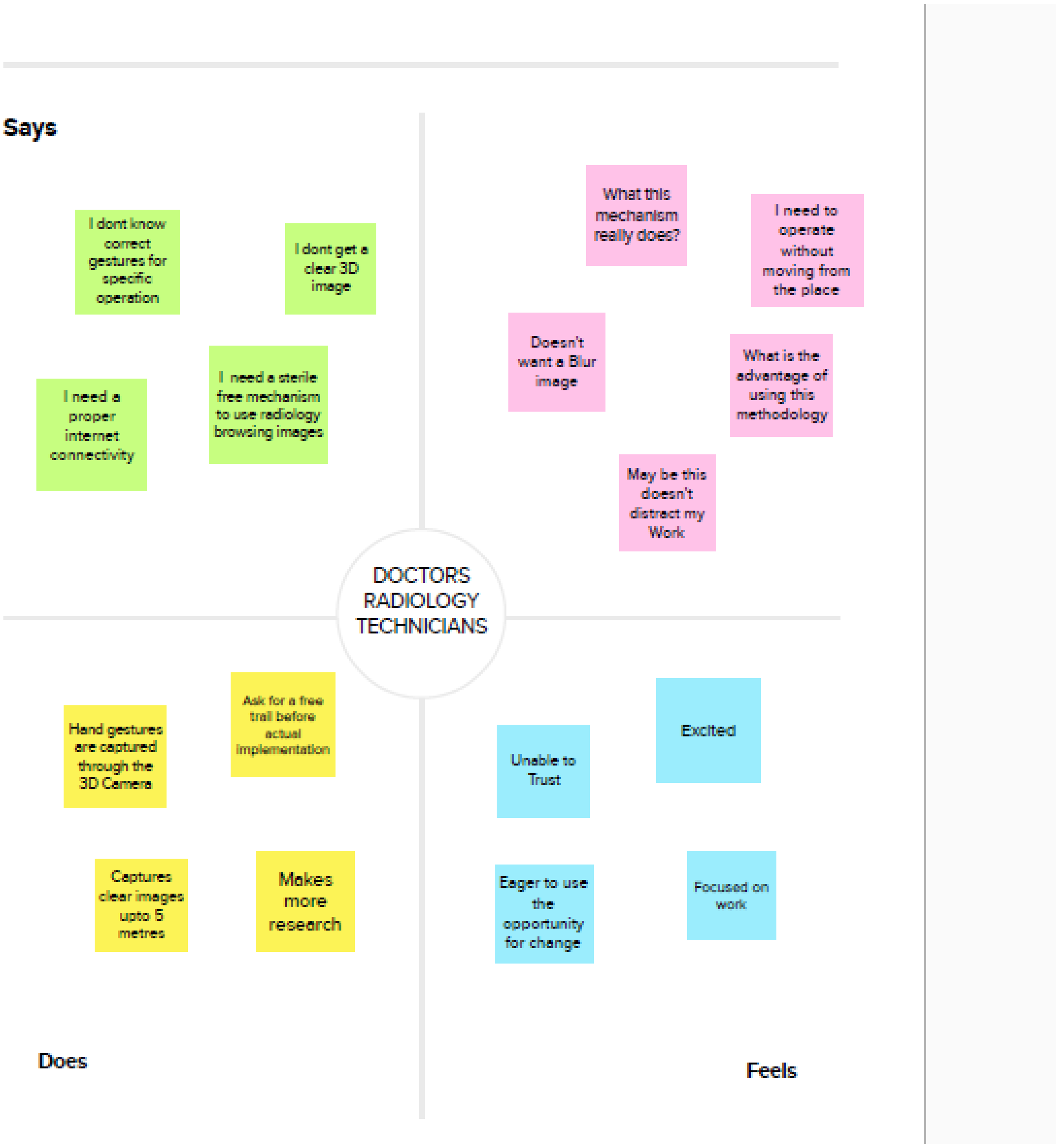
https://www.researchgate.net/publication/5401674_A_Gesturebased_Tool_for_Sterile_Browsing_of_Radiology_Images

2.3 Problem Statement Definition

Doctors have to view the images on the screen through the usage of external devices and they need to operate it. Which affects the safer means of attraction as well as leads to deviation from their work.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



Says

I dont know correct gestures for specific operation

I dont get a clear 3D image

I need a proper internet connectivity

I need a sterile free mechanism to use radiology browsing images

What this mechanism really does?

I need to operate without moving from the place

Doesn't want a Blur image

What is the advantage of using this methodology

May be this doesn't distract my Work

DOCTORS RADIOLOGY TECHNICIANS

Hand gestures are captured through the 3D Camera

Ask for a free trail before actual implementation

Captures clear images upto 5 metres

Makes more research

Unable to Trust

Excited

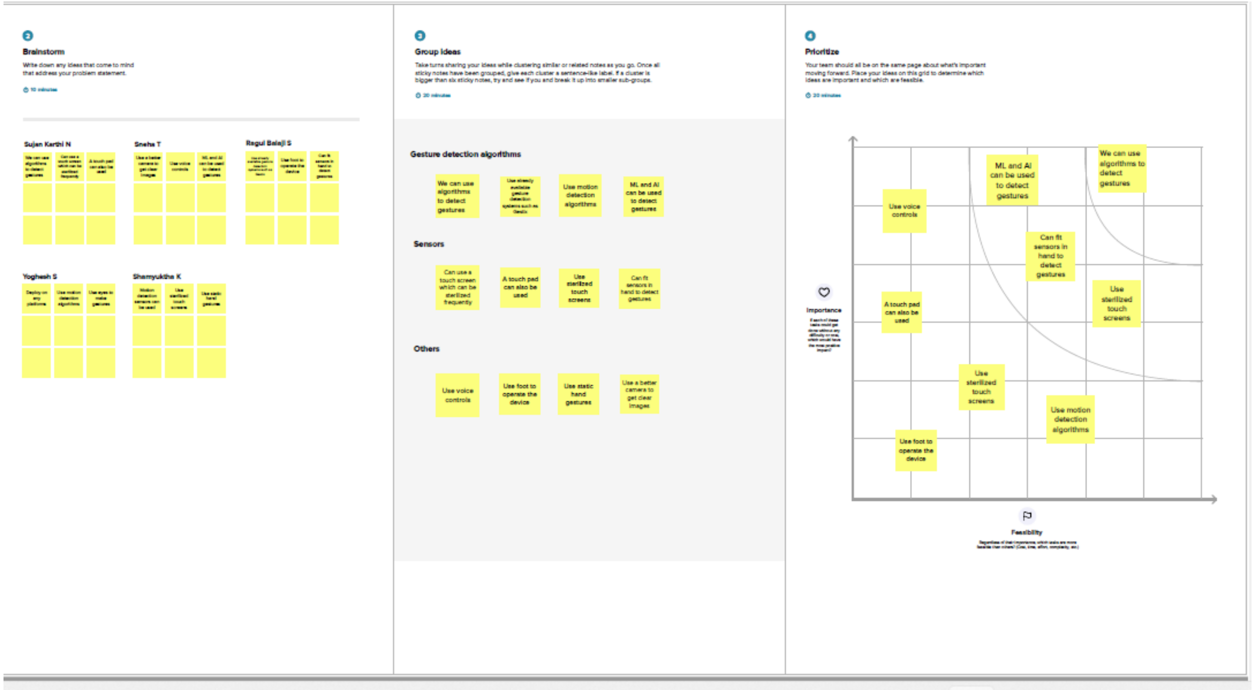
Eager to use the opportunity for change

Focused on work

Does

Feels

3.2 Ideation and Brainstorming



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Sujan Karthi N
Sneha T
Ragul Balaji S
Yoghesh S
Shamyuktha K
We can use algorithms to detect gestures
Can use a touch screen which can be sterilized frequently
A touch pad can also be used
Use a better camera to get clear images
Use voice controls
ML and AI can be used to detect gestures
Use foot to operate the device
Deploy on any platforms
Use motion detection algorithms
Use eyes to make gestures
Motion detection sensors can be used
Use sterilized touch screens
Use static hand gestures

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Gesture detection algorithms
We can use algorithms to detect gestures
Use already available gesture detection systems such as
Use motion detection algorithms
ML and AI can be used to detect gestures
Sensors
Can use a touch screen which can be sterilized frequently
A touch pad can also be used
Use sterilized touch screens
Can fit sensors in hand to detect gestures
Others
Use voice controls
Use foot to operate the device
Use static hand gestures
Use a better camera to get clear images

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

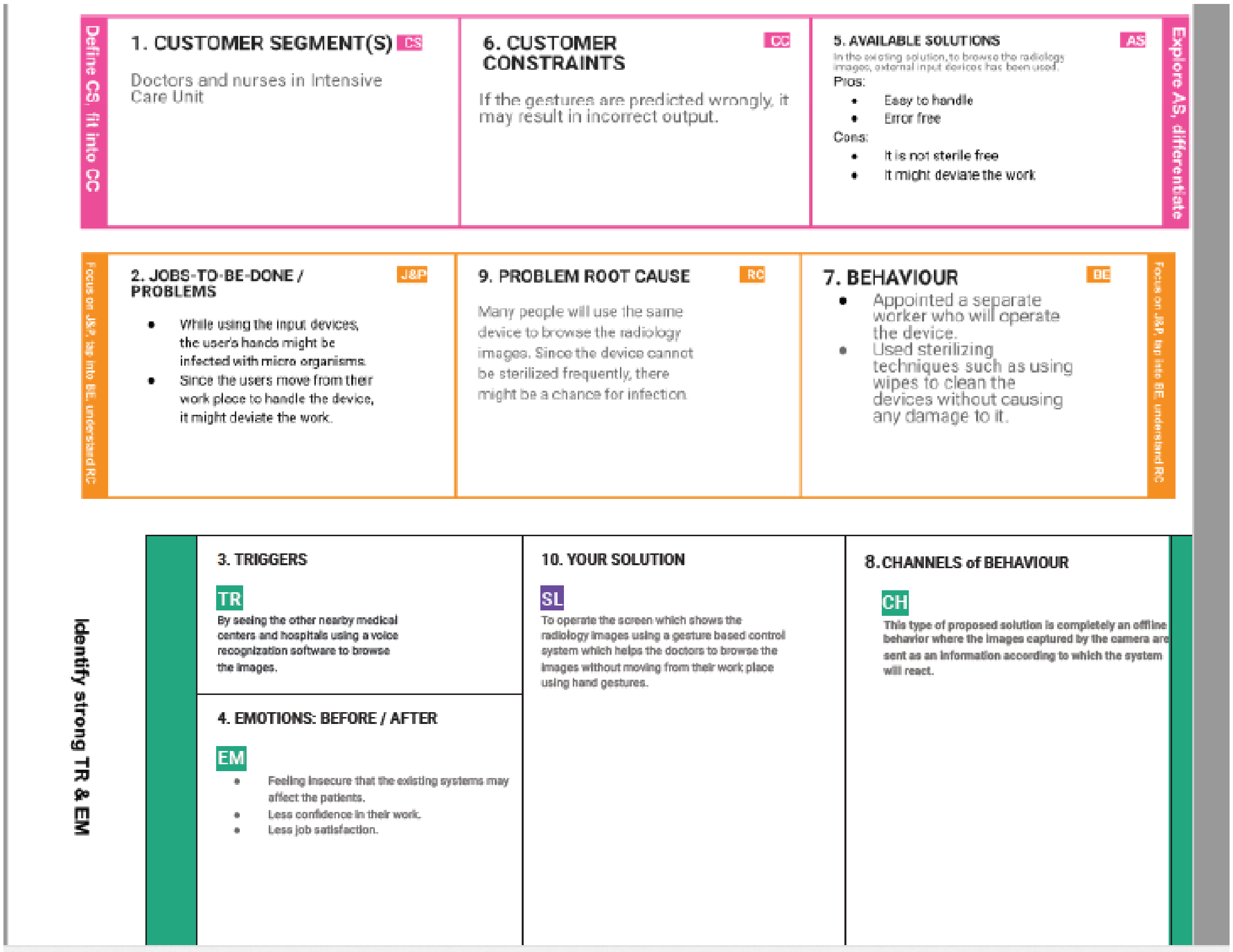
20 minutes

ML and AI can be used to detect gestures
We can use algorithms to detect gestures
Use voice controls
Can fit sensors in hand to detect gestures
A touch pad can also be used
Use sterilized touch screens
Use sterilized touch screens
Use motion detection algorithms
Use foot to operate the device
Importance
Feasibility

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To provide doctors with efficient, intuitive, accurate and safe means of interaction without affecting the quality of their work and making them work more focused.
2.	Idea / Solution description	Using gesture based tool for sterile browsing of radiology images where hand gestures are captured through camera and based on the commands by the camera the image displaying monitor reacts.
3.	Novelty / Uniqueness	Gesture based Desktop automation in which the model is pre trained on the images of different hand gestures,
4.	Social Impact / Customer Satisfaction	The doctors feels it to be more easy to operate and providing the best quality of work.
5.	Business Model (Revenue Model)	Directly contact with medical centres and hospitals to give a detail description of the system and to give a free trial to get the experience with the system.
6.	Scalability of the Solution	The proposed system will work flow by making it more reliable and focused.

3.4 Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Doctors and nurses in Intensive Care Unit

6. CUSTOMER CONSTRAINTS CC

If the gestures are predicted wrongly, it may result in incorrect output.

5. AVAILABLE SOLUTIONS AS

In the existing solution, to browse the radiology images, external input devices has been used.

Pros:

- Easy to handle
- Error free

Cons:

- It is not sterile free
- It might deviate the work

Explore AS, differentiate

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

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

- While using the input devices, the user's hands might be infected with micro organisms.
- Since the users move from their work place to handle the device, it might deviate the work.

9. PROBLEM ROOT CAUSE RC

Many people will use the same device to browse the radiology images. Since the device cannot be sterilized frequently, there might be a chance for infection.

7. BEHAVIOUR BE

- Appointed a separate worker who will operate the device.
- Used sterilizing techniques such as using wipes to clean the devices without causing any damage to it.

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

Identify strong TR & EM

3. TRIGGERS

TR

By seeing the other nearby medical centers and hospitals using a voice recognization software to browse the images.

4. EMOTIONS: BEFORE / AFTER

EM

- Feeling insecure that the existing systems may affect the patients.
- Less confidence in their work.
- Less job satisfaction.

10. YOUR SOLUTION

SL

To operate the screen which shows the radiology images using a gesture based control system which helps the doctors to browse the images without moving from their work place using hand gestures.

8. CHANNELS of BEHAVIOUR

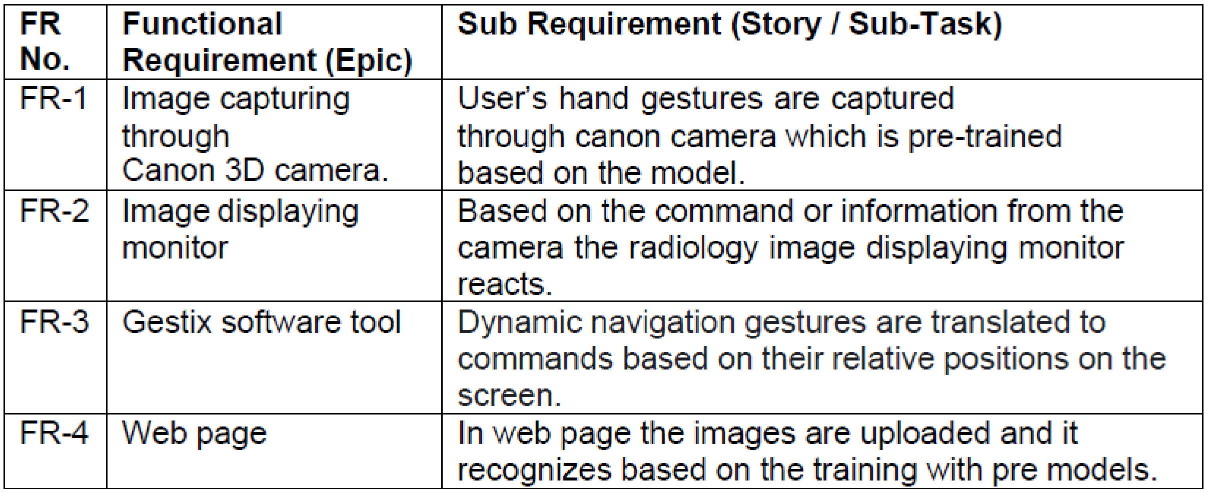
CH

This type of proposed solution is completely an offline behavior where the images captured by the camera are sent as an information according to which the system will react.

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

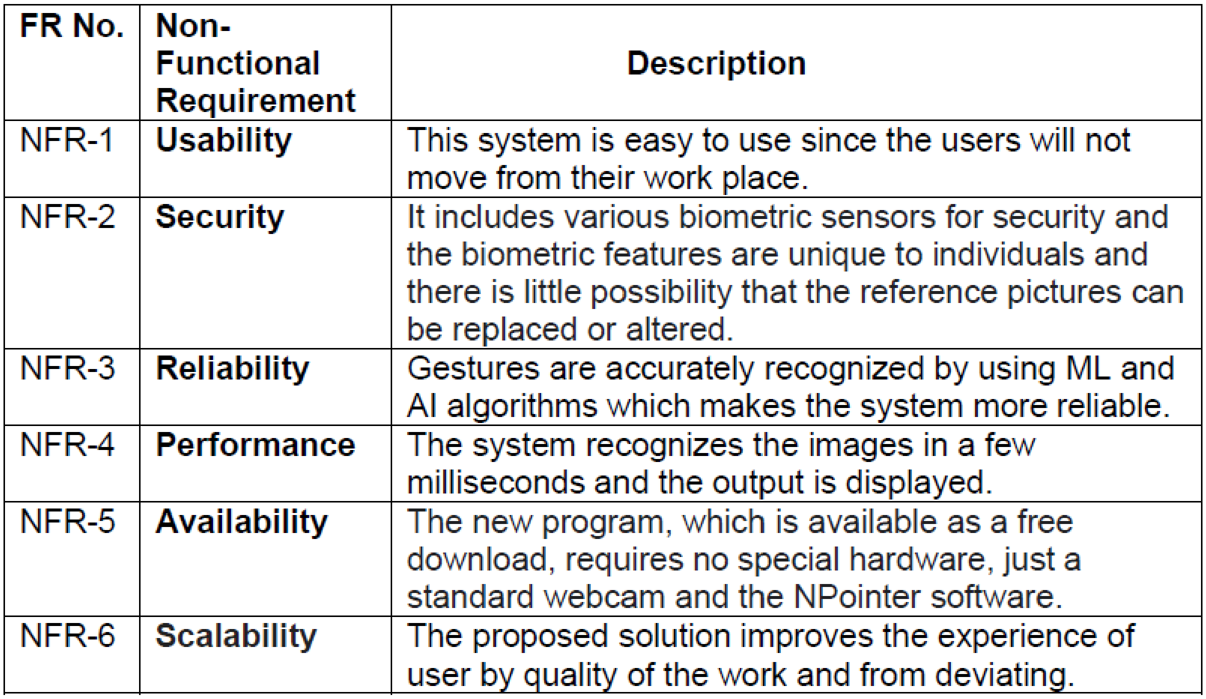
Let us see some of the functional requirements:



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Image capturing through Canon 3D camera.	User's hand gestures are captured through canon camera which is pre-trained based on the model.
FR-2	Image displaying monitor	Based on the command or information from the camera the radiology image displaying monitor reacts.
FR-3	Gestix software tool	Dynamic navigation gestures are translated to commands based on their relative positions on the screen.
FR-4	Web page	In web page the images are uploaded and it recognizes based on the training with pre models.

4.2 Non Functional Requirements

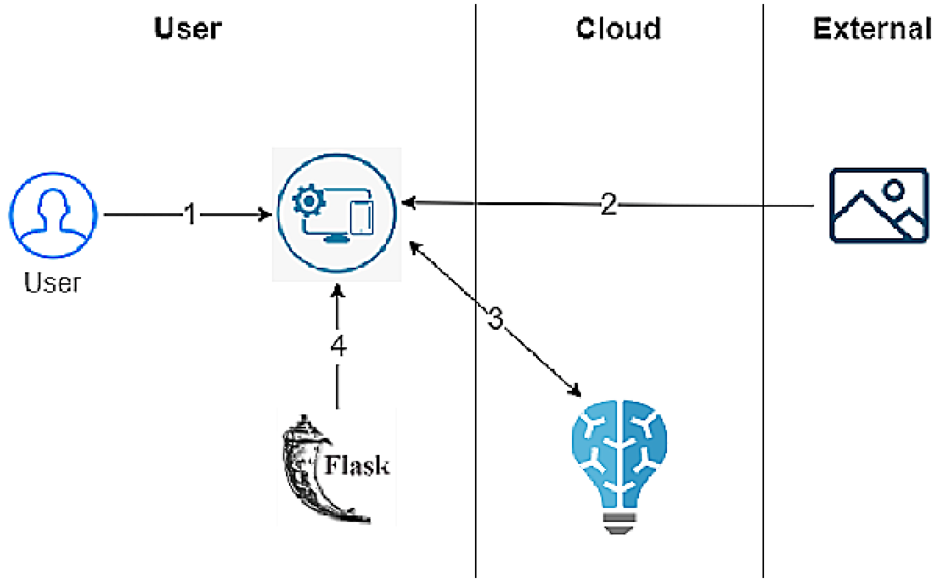
Following are the non-functional requirements of the proposed solution.



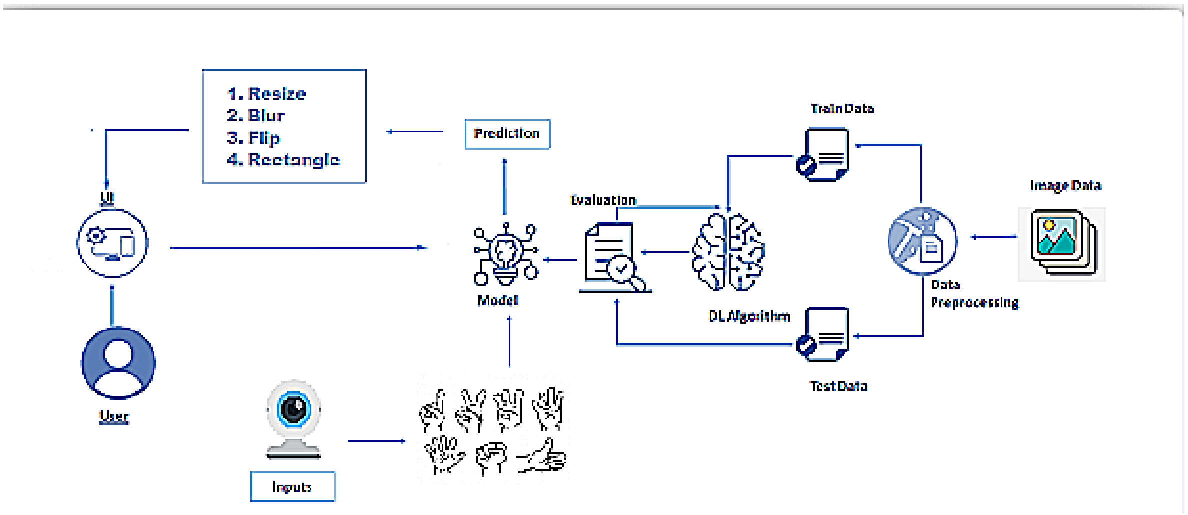
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This system is easy to use since the users will not move from their work place.
NFR-2	**Security**	It includes various biometric sensors for security and the biometric features are unique to individuals and there is little possibility that the reference pictures can be replaced or altered.
NFR-3	**Reliability**	Gestures are accurately recognized by using ML and AI algorithms which makes the system more reliable.
NFR-4	**Performance**	The system recognizes the images in a few milliseconds and the output is displayed.
NFR-5	**Availability**	The new program, which is available as a free download, requires no special hardware, just a standard webcam and the NPointer software.
NFR-6	**Scalability**	The proposed solution improves the experience of user by quality of the work and from deviating.

5.PROJECT DESIGN

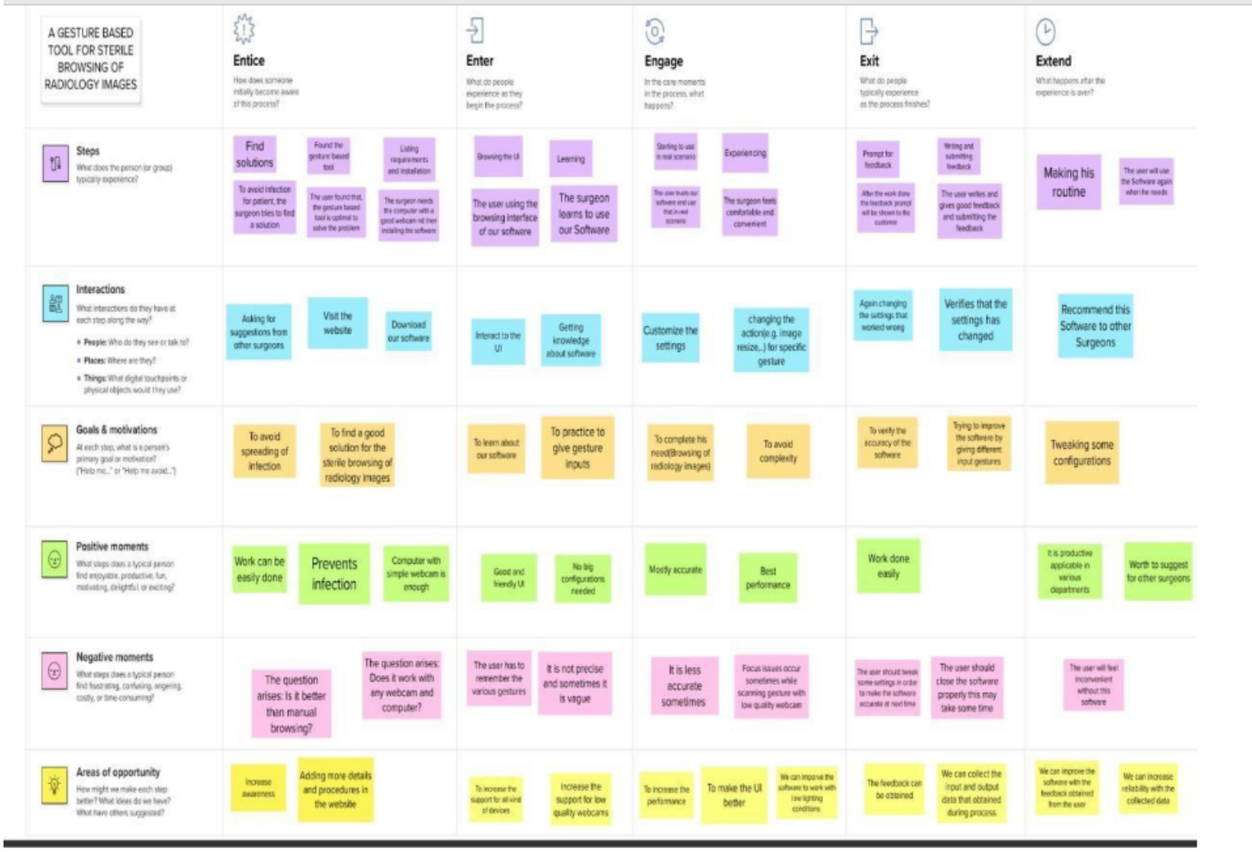
5.1 Data Flow Diagrams



5.2 Solution and Technical Architecture



5.3 User Stories



A GESTURE BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES	Entice How does someone initially become aware of this process?	Enter What do people experience as they begin the process?	Engage In the core moments in the process, what happens?	Exit What do people typically experience as the process finishes?	Extend What happens after the experience is over?
Steps What does the person (or group) typically experience?	Find solutions Found the gesture based tool Listing requirements and installation To avoid infection for patient, the surgeon tries to find a solution The user found that, the gesture based tool is optimal to solve the problem The surgeon needs the computer with a good webcam nd then installing the software	Browsing the UI Learning The user using the browsing interface of our software The surgeon learns to use our Software	Starting to use in real scenario Experiencing The user trusts our software and use that in real scenario The surgeon feels comfortable and convenient	Prompt for feedback Writing and submitting feedback After the work done the feedback prompt will be shown to the customer The user writes and gives good feedback and submitting the feedback	Making his routine The user will use the Software again when he needs
Interactions What interactions do they have at each step along the way? • People: Who do they see or talk to? • Places: Where are they? • Things: What digital touchpoints or physical objects would they use?	Asking for suggestions from other surgeons Visit the website Download our software	Interact to the UI Getting knowledge about software	Customize the settings changing the action(e.g. image resize,..) for specific gesture	Again changing the settings that worked wrong Verifies that the settings has changed	Recommend this Software to other Surgeons
Goals & motivations At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")	To avoid spreading of infection To find a good solution for the sterile browsing of radiology images	To learn about our software To practice to give gesture inputs	To complete his need(Browsing of radiology images) To avoid complexity	To verify the accuracy of the software Trying to improve the software by giving different input gestures	Tweaking some configurations
Positive moments What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?	Work can be easily done Prevents infection Computer with simple webcam is enough	Good and friendly UI No big configurations needed	Mostly accurate Best performance	Work done easily	It is productive applicable in various departments Worth to suggest for other surgeons
Negative moments What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?	The question arises: Is it better than manual browsing? The question arises: Does it work with any webcam and computer?	The user has to remember the various gestures It is not precise and sometimes it is vague	It is less accurate sometimes Focus issues occur sometimes while scanning gesture with low quality webcam	The user should tweak some settings in order to make the software accurate at next time The user should close the software properly this may take some time	The user will feel inconvenient without this software
Areas of opportunity How might we make each step better? What ideas do we have? What have others suggested?	Increase awareness Adding more details and procedures in the website	To increase the support for all kind of devices Increase the support for low quality webcams	To increase the performance To make the UI better We can improve the software to work with low lighting conditions	The feedback can be obtained We can collect the input and output data that obtained during process	We can improve the software with the feedback obtained from the user We can increase reliability with the collected data

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning &Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	10 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	13 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and Prioritize the top 3 ideas based on the feasibility & importance.	13 SEPTEMBER 2022

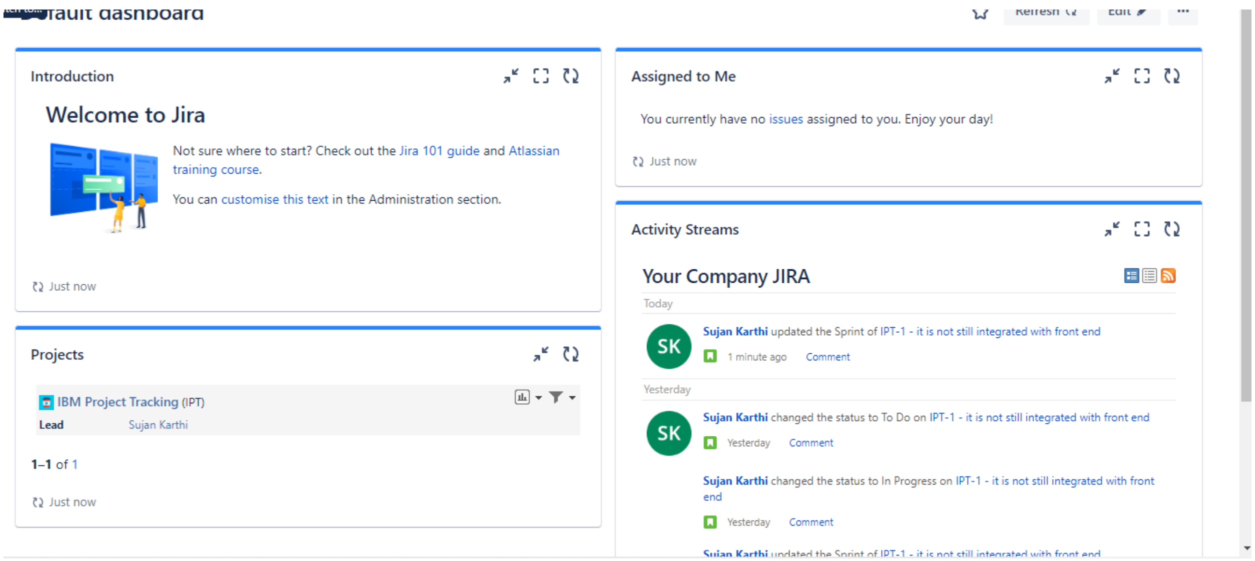
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	27 SEPTEMBER 2022
Prpblem Solution Fit	Prepare problem - solution Fit document.	27 SEPTEMBER 2022
Solution Architecture	Prepare solution Architecture document.	10 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	14 OCTOBER 2022
Data Flow Diagrams	Draw the data flow Diagrams and submit for review.	18 OCTOBER 2022
Technology Architecture	Architecture diagram.	24 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & Activity list of the project.	27 OCTOBER 2022
Project Development - Delivery of Sprint 1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS….

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Download the Dataset	5	High	Yoghesh S
Sprint-1		USN-2	Image Pre-processing	5	High	Sujan Karthi N Sneha T
Sprint-1		USN-3	Import and Configure the Image Data Generator Library and Class	5	High	RagulBalaji

Sprint-1		USN-4	Apply Image Data Generator Functionality to Train-Set and Test-Set	5	High	Shamyuktha K
Sprint-2	Model Building	USN-5	Import the Model Building Libraries and Initializing the Model	5	High	Sujan Karthi N
Sprint-2		USN-6	Adding CNN Layers and Dense Layers	5	High	Yoghesh S
Sprint-2		USN-7	Configure the Learning Process	5	High	Sneha T Shamyuktha
Sprint-2		USN-8	Train the Model, Save the Model and Test the Model	3	Low	RagulBalaji S Sujan Karthi N
Sprint-3	Application Building	USN-9	Create Web Application using HTML, CSS, JavaScript	10	High	Sneha T Shamyuktha K Yoghesh S Ragul Balaji S Sujan karthi N

6.3 Reports from Jira



fault dashboard

Introduction

Welcome to Jira

Not sure where to start? Check out the Jira 101 guide and Atlassian training course.

You can customise this text in the Administration section.

Just now

Projects

IBM Project Tracking (IPT)

Lead	Sujan Karthi

1–1 of 1

Just now

Assigned to Me

You currently have no issues assigned to you. Enjoy your day!

Just now

Activity Streams

Your Company JIRA

Today

SK Sujan Karthi updated the Sprint of IPT-1 - it is not still integrated with front end

1 minute ago Comment

Yesterday

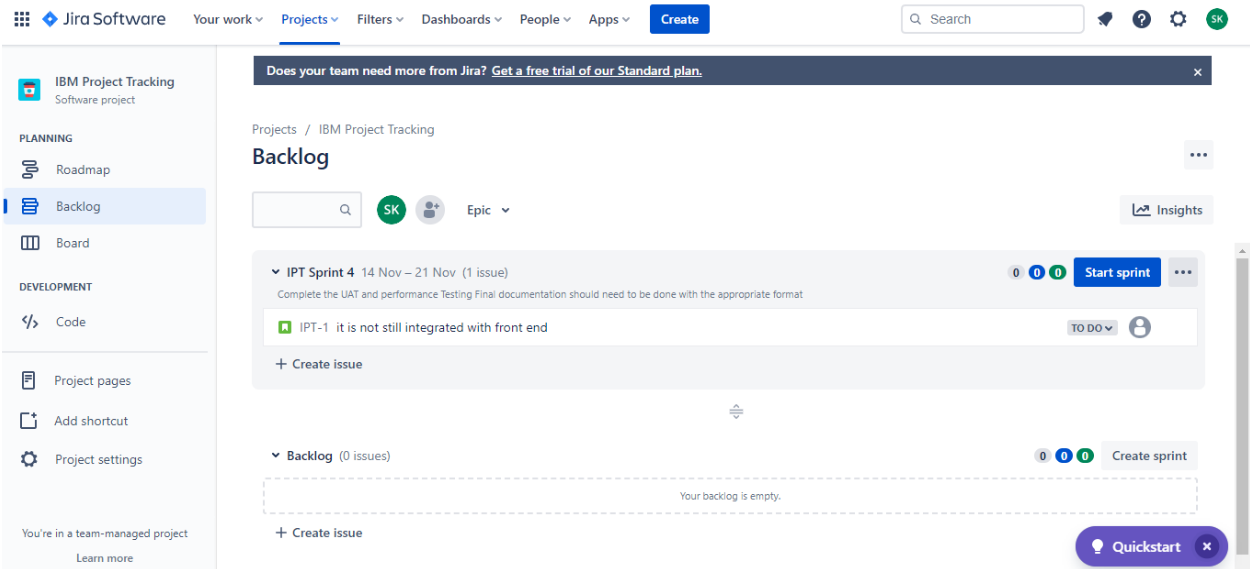
SK Sujan Karthi changed the status to To Do on IPT-1 - it is not still integrated with front end

Yesterday Comment

Sujan Karthi changed the status to In Progress on IPT-1 - it is not still integrated with front end

Yesterday Comment

Sujan Karthi updated the Sprint of IPT-1 - it is not still integrated with front end



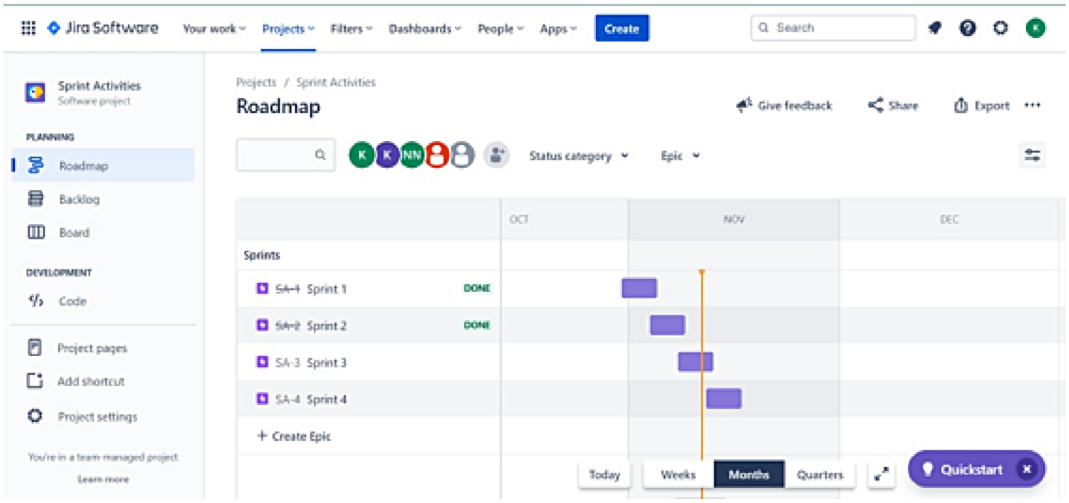
IPT Sprint 4 14 Nov – 21 Nov (1 issue) 0 0 0

Complete the UAT and performance Testing Final documentation should need to be done with the appropriate format

IPT-1 it is not still integrated with front end TO DO

Backlog (0 issues) 0 0 0

ROADMAP

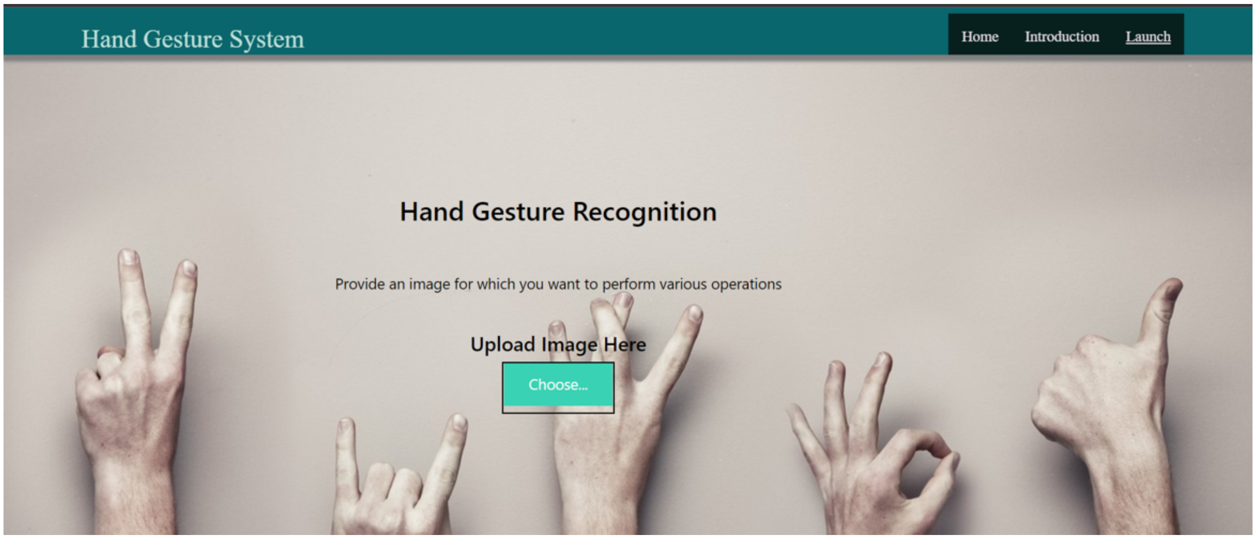


~~SA-1~~	Sprint 1	DONE
~~SA-2~~	Sprint 2	DONE
SA-3	Sprint 3	
SA-4	Sprint 4	

The tasks are assigned to everyone for every sprint and completed the tasks successfully.

7. FEATURES

7.1 Feature 1


Hand Gesture System

Hand Gesture Recognition

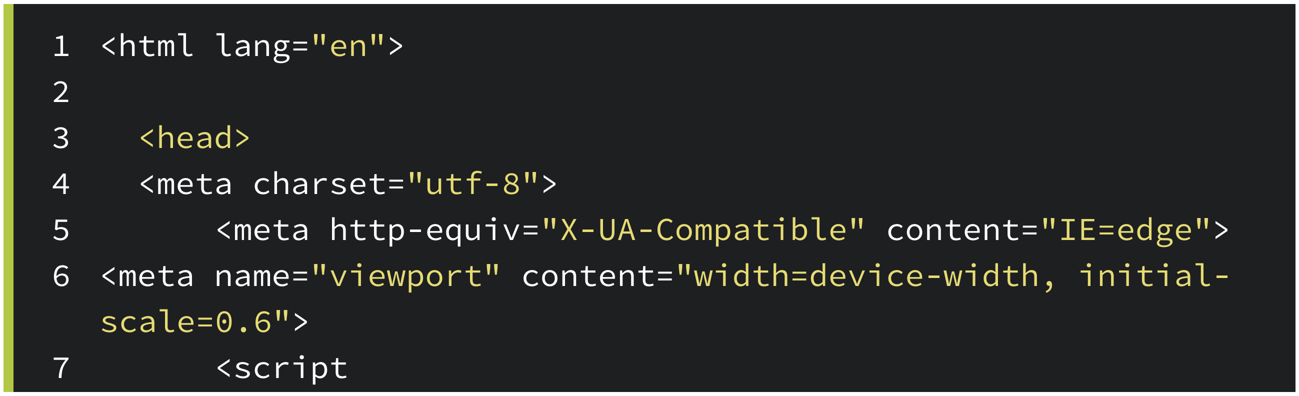
Provide an image for which you want to perform various operations

Upload Image Here

This is the most important and notable feature of our project where we can choose the images for hand gesture recognition.

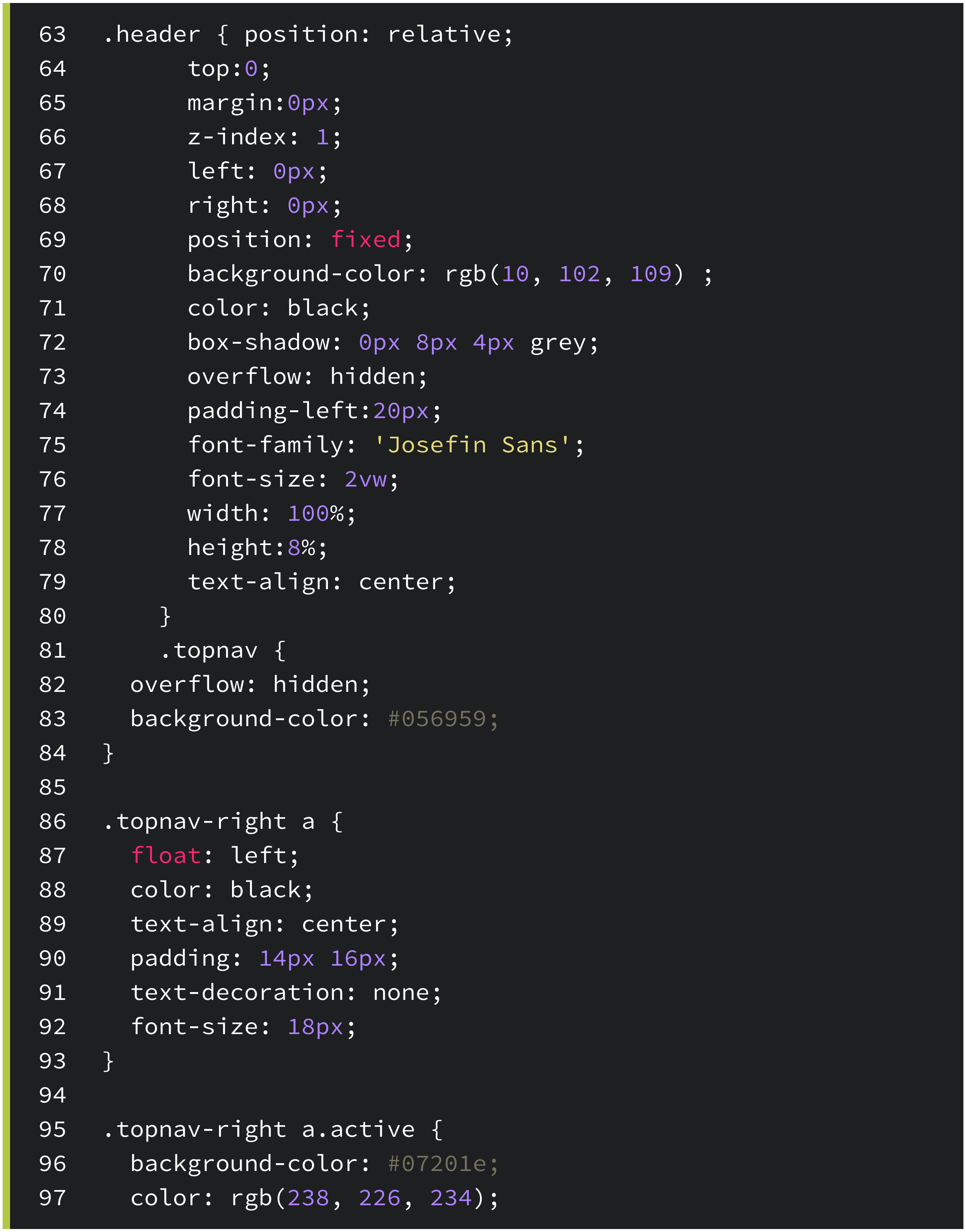
CODE :

Here we are inserting a sample of our code for this particular feature.



```
<html lang="en">

  <head>
  <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=0.6">
      <script
```

```
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

      <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
      <meta charset="UTF-8">
      <title>Predict</title>
      <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">

      <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
      <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
      <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
      <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
  <style>
  .bar
  {
  margin: 0px;
  padding:20px;
  background-color:black;
  opacity:0.6;
  color:black;
  font-family:'Roboto',sans-serif;
  font-style: italic;
  border-radius:20px;
  font-size:25px;
  }
  a
  {
  color:black;
  float:right;
  text-decoration:none;
  font-style:normal;
  padding-right:20px;
  }
  a:hover{
  background-color:black;
  color:black;
  font-size:30px;
  padding-left:10px;
  }

  div1{
    text-align: center;
    width: 650spx;
    height: 800px;
    padding: 190px;
    margin: 10px;
    position: absolute;
  }



  body
  {
      background-image: url("https://img3.goodfon.com/wallpaper/nbig/a/af/ruki-znaki-steny.jpg");
      background-size: cover;
  }

```



```css
.header { position: relative;
      top:0;
      margin:0px;
      z-index: 1;
      left: 0px;
      right: 0px;
      position: fixed;
      background-color: rgb(10, 102, 109) ;
      color: black;
      box-shadow: 0px 8px 4px grey;
      overflow: hidden;
      padding-left:20px;
      font-family: 'Josefin Sans';
      font-size: 2vw;
      width: 100%;
      height:8%;
      text-align: center;
    }
    .topnav {
  overflow: hidden;
  background-color: #056959;
}

.topnav-right a {
  float: left;
  color: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

.topnav-right a.active {
  background-color: #07201e;
  color: rgb(238, 226, 234);
}

                      .topnav-right a:hover {
                        background-color: rgb(181, 228, 236);
                        color: rgb(6, 27, 36);
                      }



                      .topnav-right {
                        float: right;
                        padding-right:100px;
                      }
                    .button {
                      background-color: #091425;
                      border: none;
                      color: black;
                      padding: 15px 32px;
                      text-align: center;
                      text-decoration: none;
                      display: inline-block;
                      font-size: 12px;
                      border-radius: 16px;
                      }
                      .button:hover {
                        box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
                      }
                      form {border: 2px solid black; margin-left:400px;margin-right:400px;}

                      input[type=text], input[type=password] {
                        width: 100%;
padding: 12px 20px;
display: inline-block;
margin-bottom:18px;
border: 1px solid #ccc;
box-sizing: border-box;
}

button {
background-color: #091425;
color: black;
padding: 14px 20px;
margin-bottom:10px;
border: none;
cursor: pointer;
width: 17%;
border-radius:4px;
font-family:Montserrat;
}

button:hover {
opacity: 0.8;
}

.cancelbtn {
width: auto;
padding: 10px 18px;
background-color: #f44336;
}

.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
}

img.avatar {
```

```css
  width: 30%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}

.home{
  margin:80px;

  width: 84%;
  height: 500px;
  padding-top:10px;
  padding-left: 30px;

}
```

```
.login{
  margin:80px;
  box-sizing: content-box;
  width: 84%;
  height: 420px;
  padding: 30px;
  border: 10px solid rgb(12, 91, 94);
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
 border: 10px solid rgb(12, 91, 94);
}

.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
```

```
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

.active {
  background-color: #267481;
}

/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text
```

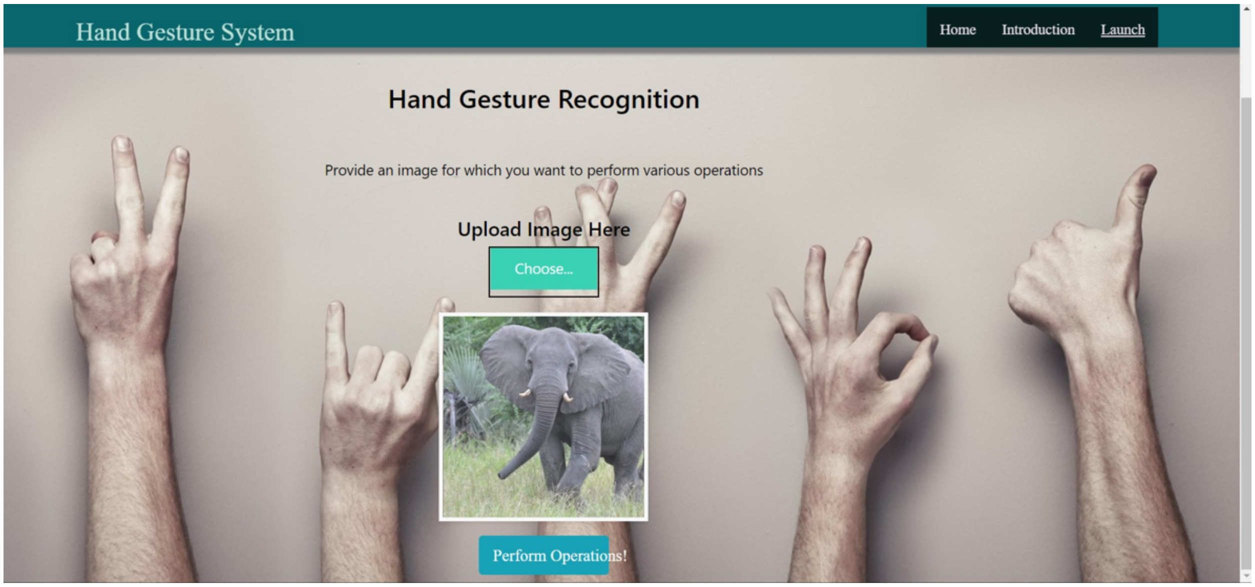
```
size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}

</style>
</head>

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

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

  <div1 styles=""><h1><font color="Black" size="6" font-family="Roboto">Hand Gesture Recognition</h1><br>
  <p><i><font color="Black" size="4" fonr-family="sans-serif"></i>Provide an image for which you want to perform various operations</p>
  <br>
```

```html
    <div>
      <h4>Upload Image Here</h4>
  <form action = "http://localhost:5000/" id="upload-file" method="post" enctype="multipart/form-data">
    <label for="imageUpload" class="upload-label">
      Choose...
    </label>
    <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg,.pdf">
  </form>
        <center>
  <div class="image-section" style="display:none;">
    <div class="img-preview">
      <div id="imagePreview">
      </div>
    </div>
    <div>
      <button type="button" class="btn btn-info btn-lg " id="btn-predict">Predict!</button>
    </div>
  </div>
  <div class="loader" style="display:none;"></div>
        </center>
</div>
</div1>

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

```
</footer>

</html>
```

7.2 FEATURE 2

After selecting the images we will upload the images after that with the help of the trained model and developed code using flask and Open CV we can predict our results.



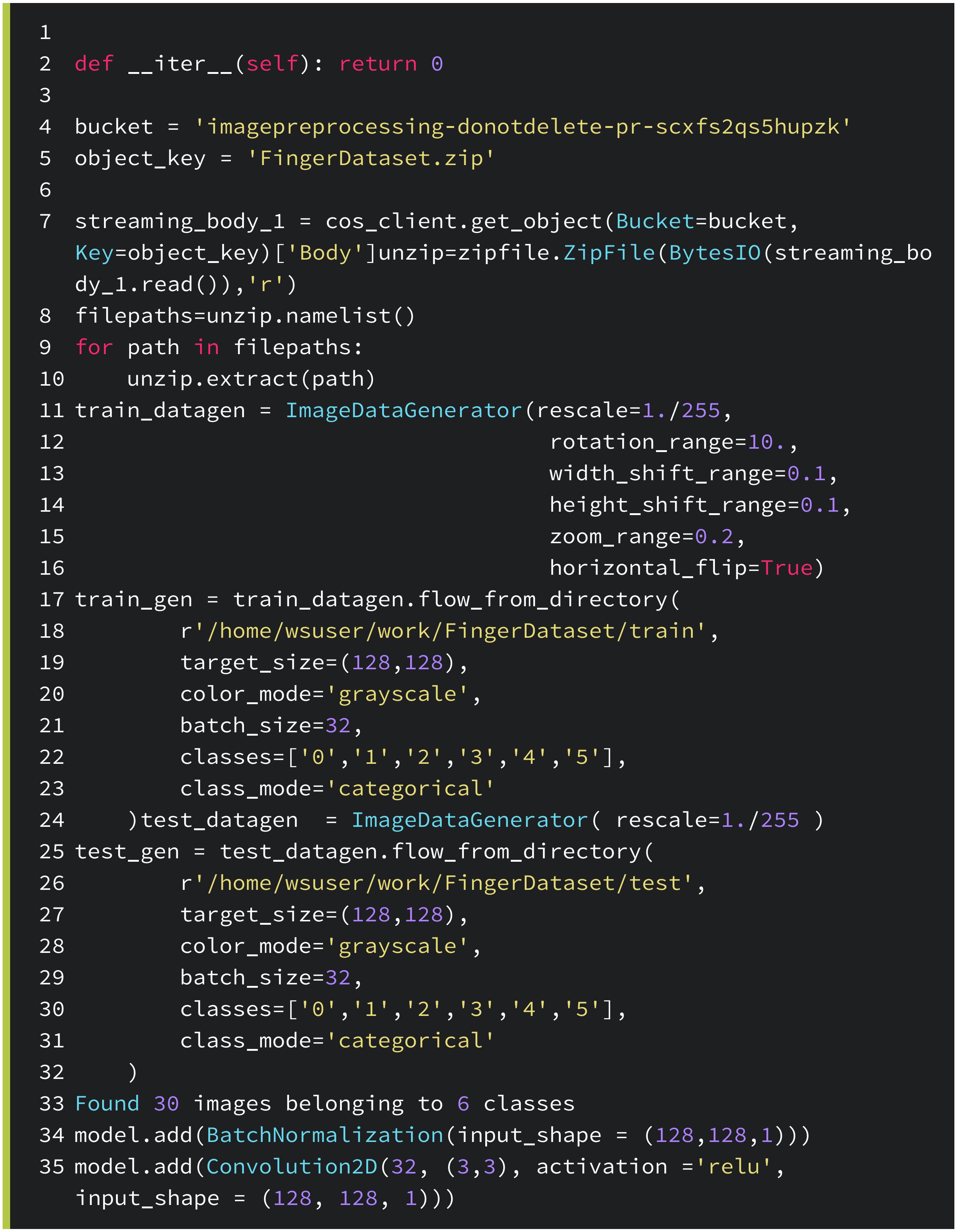
Hand Gesture System

Hand Gesture Recognition

Provide an image for which you want to perform various operations

Upload Image Here

```
import os, types
from ibm_watson_machine_learning import APIClient
import pandas as pd
from botocore.client import Config
import ibm_boto3
from tensorflow.keras.models import Sequential
```



```

def __iter__(self): return 0

bucket = 'imagepreprocessing-donotdelete-pr-scxfs2qs5hupzk'
object_key = 'FingerDataset.zip'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
filepaths=unzip.namelist()
for path in filepaths:
    unzip.extract(path)
train_datagen = ImageDataGenerator(rescale=1./255,
                                    rotation_range=10.,
                                    width_shift_range=0.1,
                                    height_shift_range=0.1,
                                    zoom_range=0.2,
                                    horizontal_flip=True)
train_gen = train_datagen.flow_from_directory(
        r'/home/wsuser/work/FingerDataset/train',
        target_size=(128,128),
        color_mode='grayscale',
        batch_size=32,
        classes=['0','1','2','3','4','5'],
        class_mode='categorical'
    )test_datagen  = ImageDataGenerator( rescale=1./255 )
test_gen = test_datagen.flow_from_directory(
        r'/home/wsuser/work/FingerDataset/test',
        target_size=(128,128),
        color_mode='grayscale',
        batch_size=32,
        classes=['0','1','2','3','4','5'],
        class_mode='categorical'
    )
Found 30 images belonging to 6 classes
model.add(BatchNormalization(input_shape = (128,128,1)))
model.add(Convolution2D(32, (3,3), activation ='relu', input_shape = (128, 128, 1)))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=6,kernel_size=4,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=3,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=2,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Flatten())
model.add(Dense(units=128,activation = 'relu'))
model.add(Dense(units = 64, activation = 'relu'))
model.add(Dense(units = 32, activation = 'relu'))
model.add(Dense(units = 6, activation = 'softmax'))
```

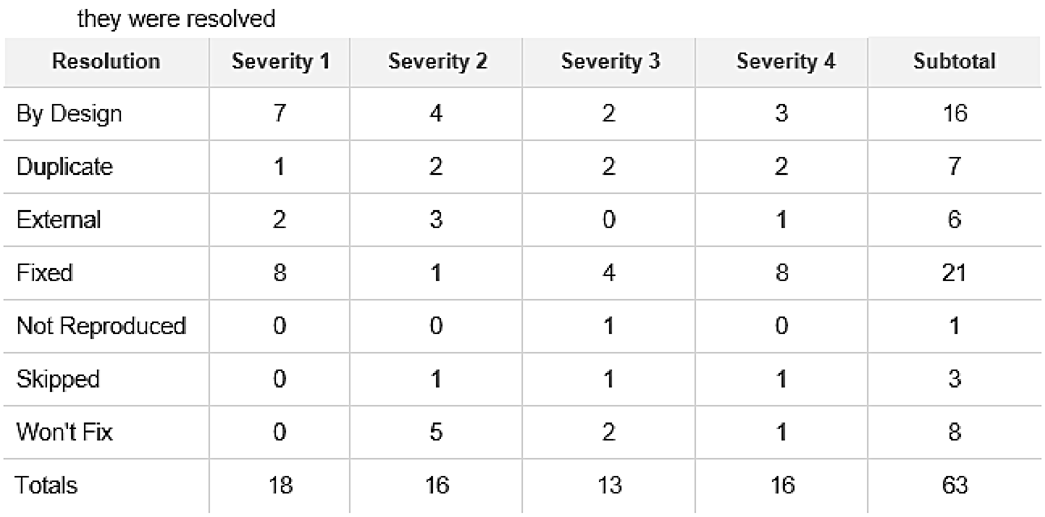
8 TESTING

8.1 Test Cases

Test Scenarios
Navigation Bar
Verify the title in the Navigator Bars
Validate all the tabs in the Navigator bars
Verify the User is not redirected to the wrong page
Home Page
Verify the visibility of the video should be true in Homepage
Validate the description of the video in the home page
Verify the user is able to Navigate to the introduction page
Introduction Page
Verify the user is in the introduction page
Verify the page title and introduction description.
Launch Page
Verify the User is in the launch page
Verify the upload image option in the launch page
Verify the choose button is enabled

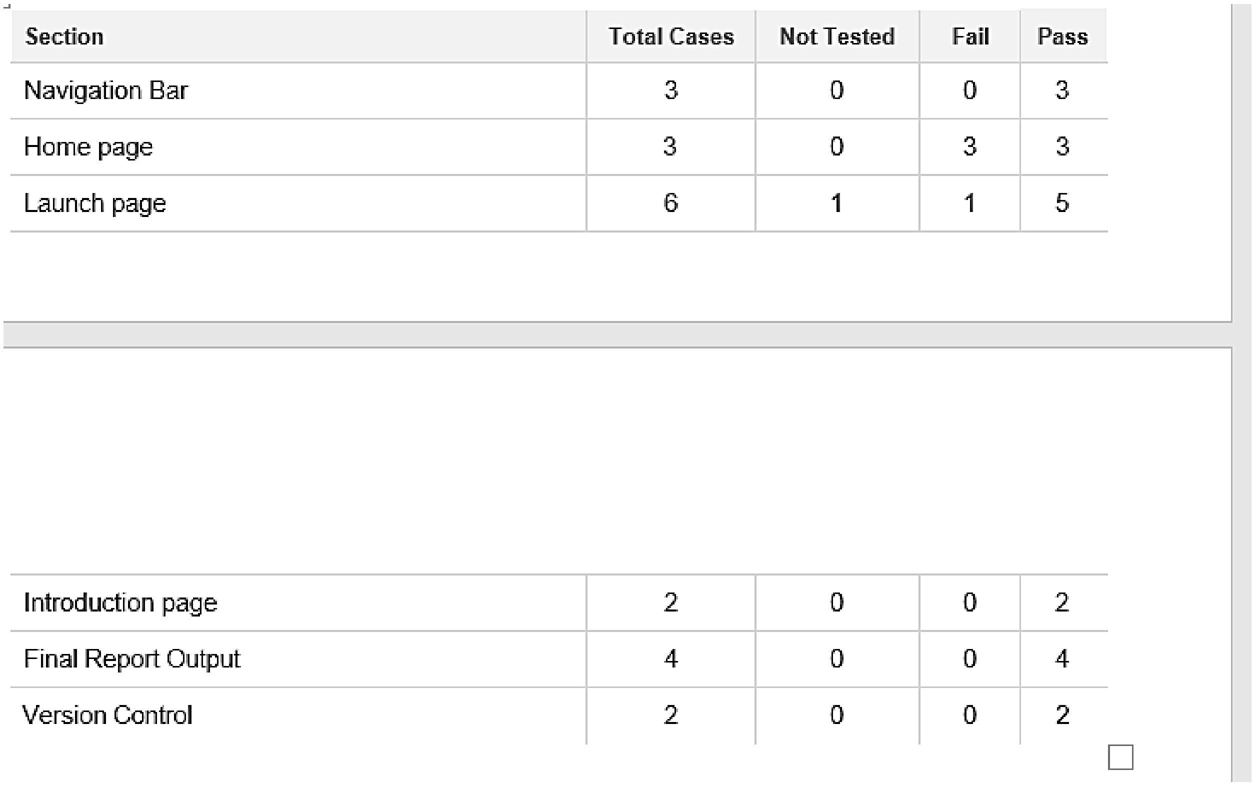
Verify the user is able to access the files from their PC
Verify the user is unable to upload no files.
Validate the selected image is same as uploaded image

8.2 USER ACCEPTANCE TESTING



they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	4	2	3	16
Duplicate	1	2	2	2	7
External	2	3	0	1	6
Fixed	8	1	4	8	21
Not Reproduced	0	0	1	0	1
Skipped	0	1	1	1	3
Won't Fix	0	5	2	1	8
Totals	18	16	13	16	63



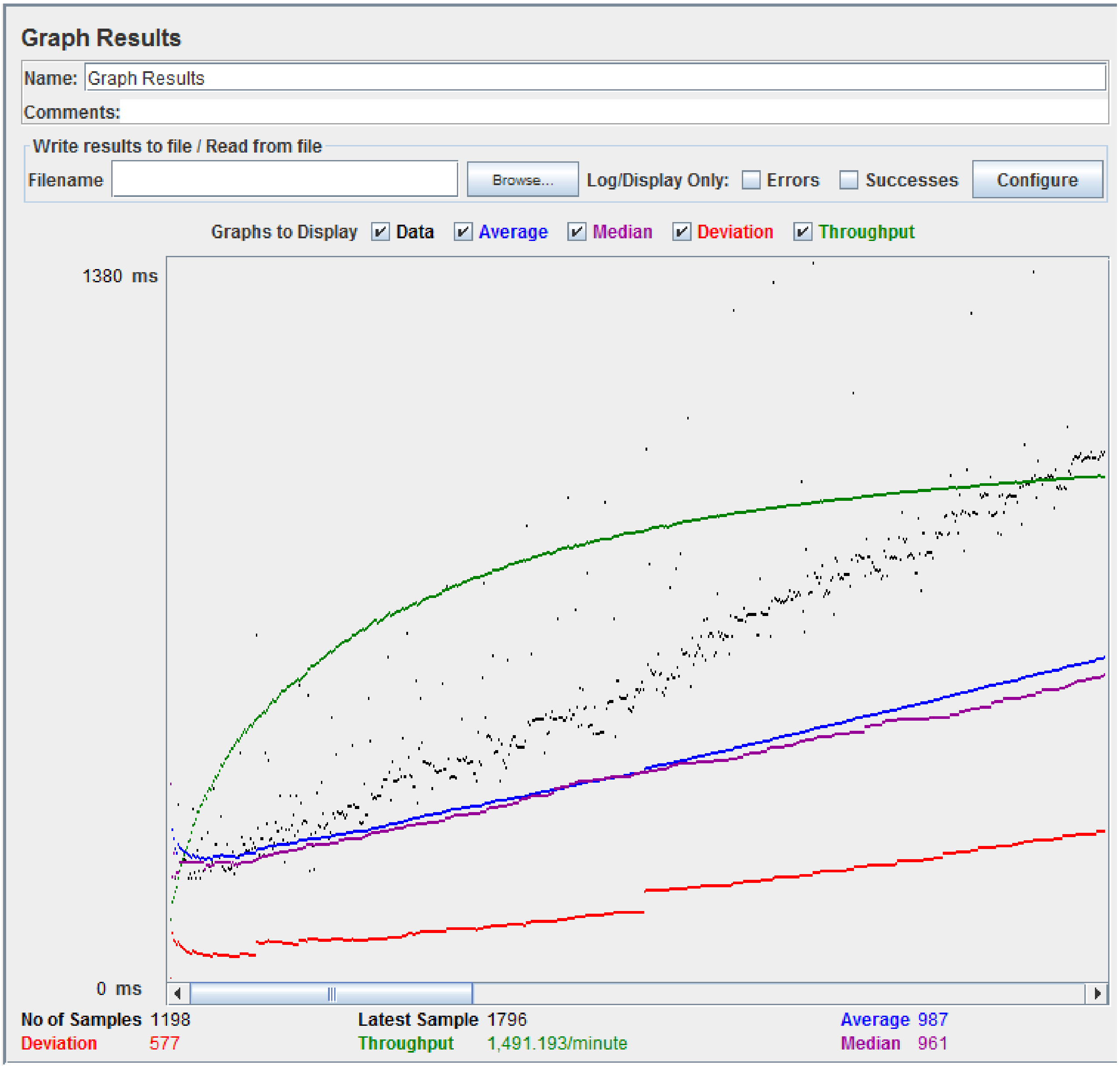
Section	Total Cases	Not Tested	Fail	Pass
Navigation Bar	3	0	0	3
Home page	3	0	3	3
Launch page	6	1	1	5
Introduction page	2	0	0	2
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.RESULTS

9.1.Performance Metrics

JMeter Load Testing:

Load graph:



Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

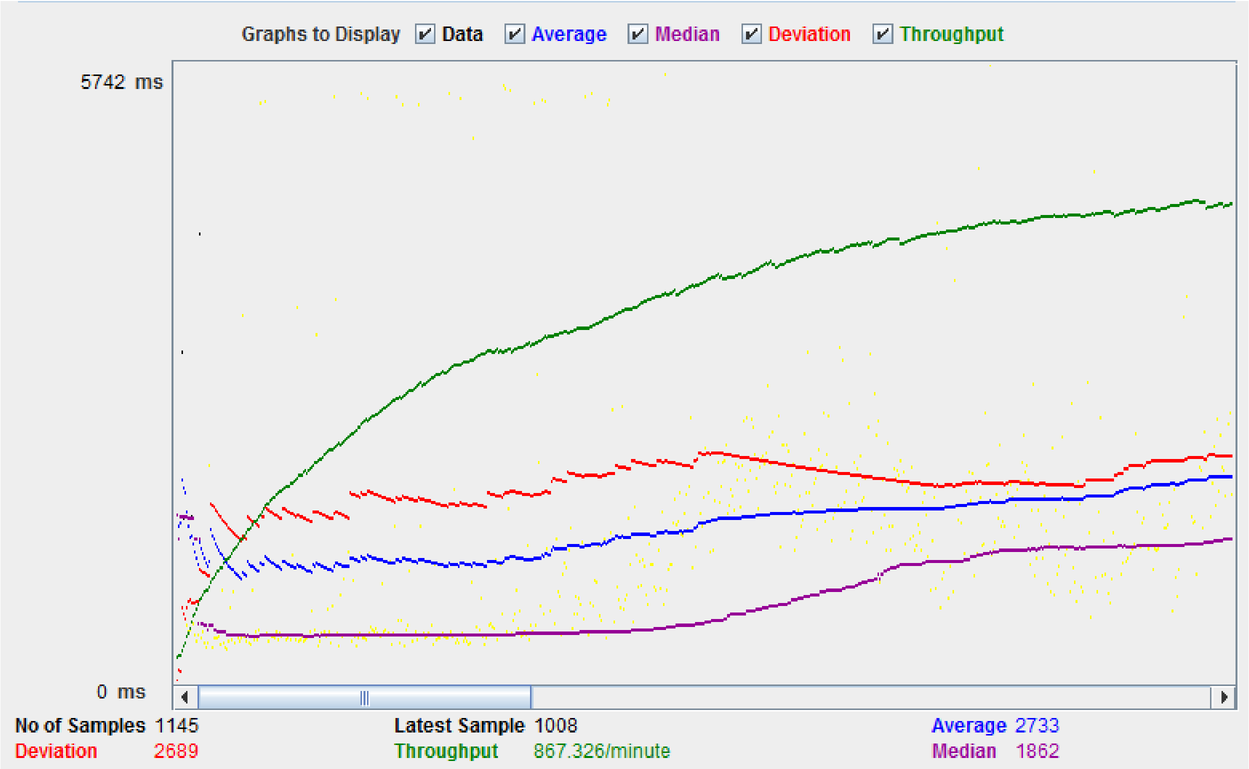
No of Samples 1198
Deviation 577

Latest Sample 1796
Throughput 1,491.193/minute

Average 987
Median 961

JMeter Performance Testing

Performance graph:



Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 1145
Deviation 2689
Latest Sample 1008
Throughput 867.326/minute
Average 2733
Median 1862

10.ADVANTAGES AND 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

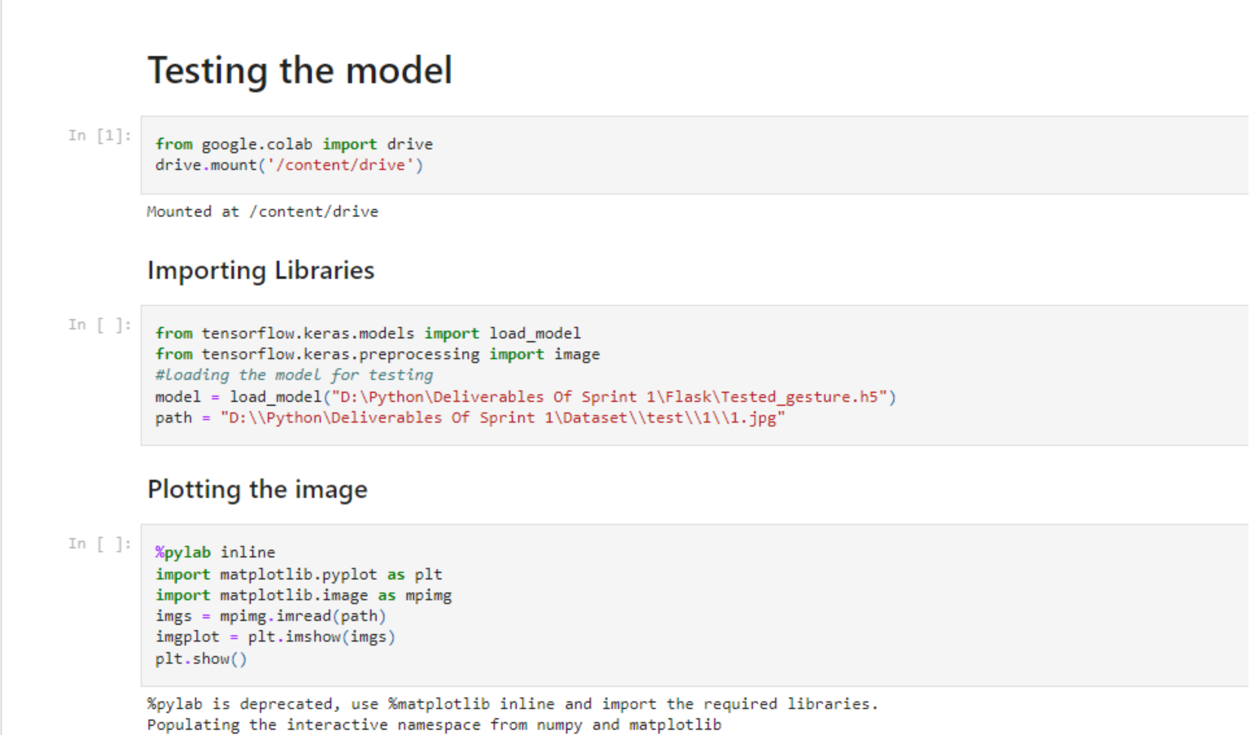
In this project we developed a tool which recognises hand gestures and enables doctors to browse through radiology images using these gestures. This enables doctors and surgeons to maintain the sterility as they would not have to touch any mouse or keyboard to go through the images. This tool is also 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 ofgestures 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



Testing the model

In [1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

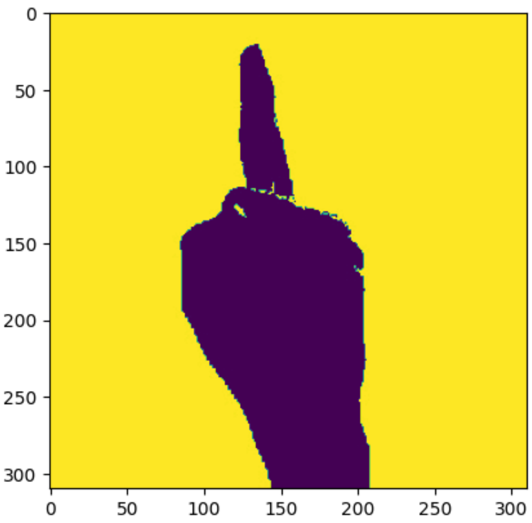
Importing Libraries

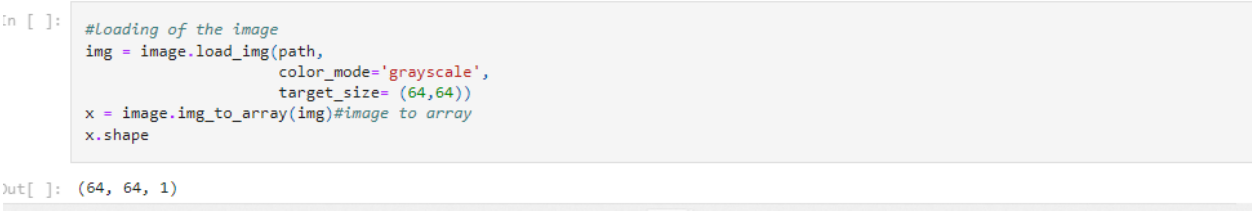
In [ ]:
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
#Loading the model for testing
model = load_model("D:\Python\Deliverables Of Sprint 1\Flask\Tested_gesture.h5")
path = "D:\\Python\Deliverables Of Sprint 1\Dataset\\test\\1\\1.jpg"
```

Plotting the image

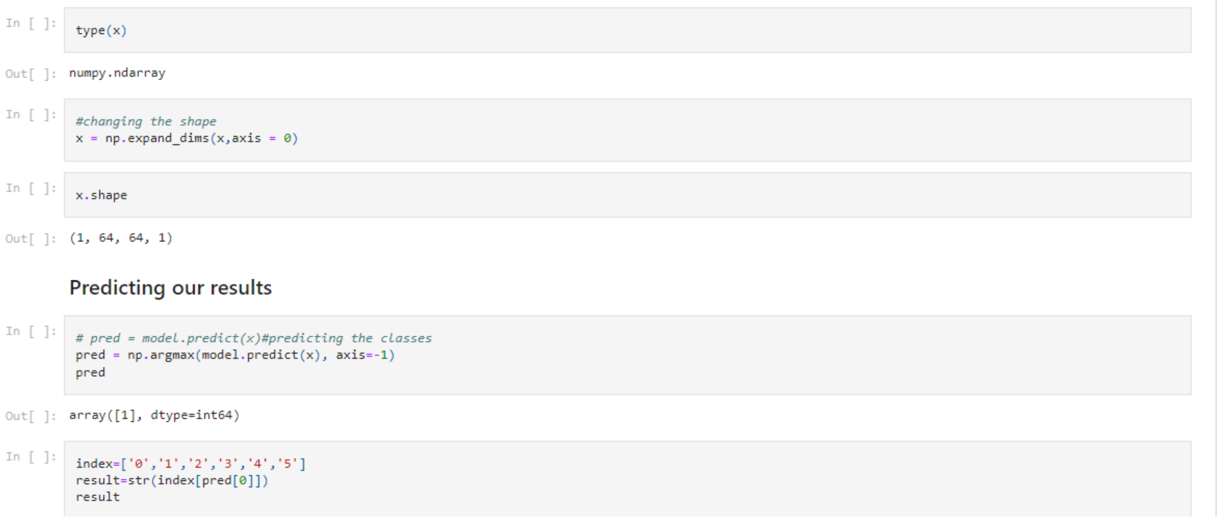
In [ ]:
```
%pylab inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
imgs = mpimg.imread(path)
imgplot = plt.imshow(imgs)
plt.show()
```

```
%pylab is deprecated, use %matplotlib inline and import the required libraries.
Populating the interactive namespace from numpy and matplotlib
```





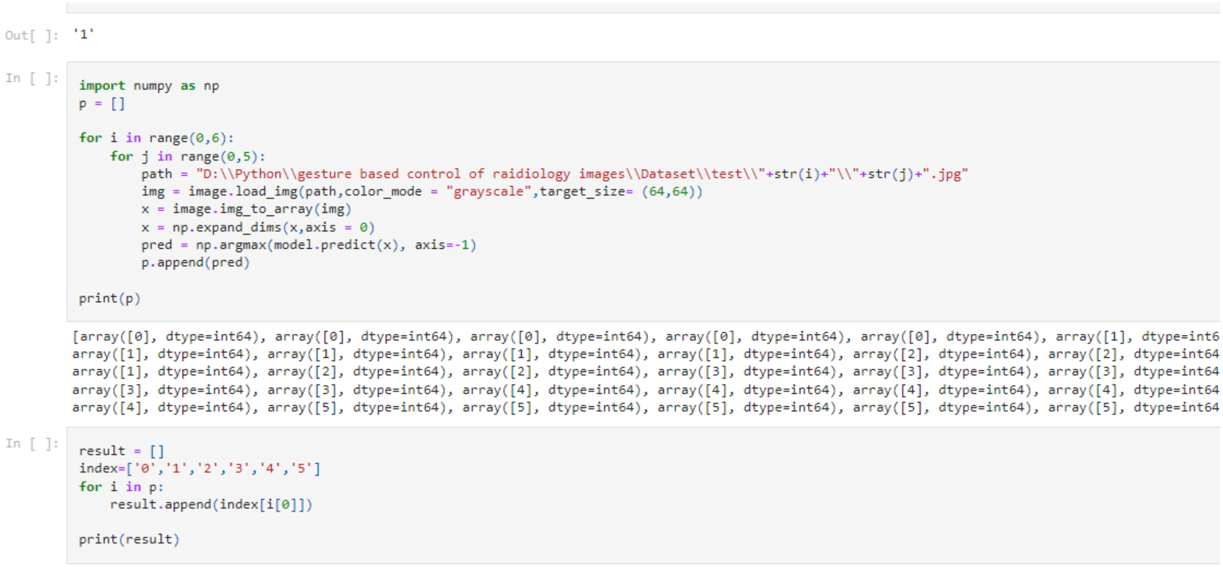
```
In [ ]: #loading of the image
        img = image.load_img(path,
                             color_mode='grayscale',
                             target_size= (64,64))
        x = image.img_to_array(img)#image to array
        x.shape

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



```
In [ ]: type(x)

Out[ ]: numpy.ndarray

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

In [ ]: x.shape

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

Predicting our results

```
In [ ]: # pred = model.predict(x)#predicting the classes
        pred = np.argmax(model.predict(x), axis=-1)
        pred

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

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

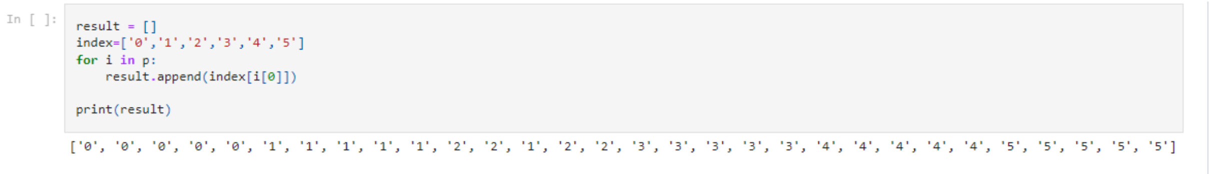


```
Out[ ]: '1'
```

```
In [ ]: import numpy as np
        p = []

        for i in range(0,6):
            for j in range(0,5):
                path = "D:\\Python\\gesture based control of raidiology images\\Dataset\\test\\"+str(i)+"\\"+str(j)+".jpg"
                img = image.load_img(path,color_mode = "grayscale",target_size= (64,64))
                x = image.img_to_array(img)
                x = np.expand_dims(x,axis = 0)
                pred = np.argmax(model.predict(x), axis=-1)
                p.append(pred)

        print(p)
```

```
[array([0], dtype=int64), array([0], dtype=int64), array([0], dtype=int64), array([0], dtype=int64), array([0], dtype=int64), array([1], dtype=int6
array([1], dtype=int64), array([1], dtype=int64), array([1], dtype=int64), array([1], dtype=int64), array([2], dtype=int64), array([2], dtype=int64
array([1], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), array([3], dtype=int64), array([3], dtype=int64), array([3], dtype=int64
array([3], dtype=int64), array([3], dtype=int64), array([4], dtype=int64), array([4], dtype=int64), array([4], dtype=int64), array([4], dtype=int64
array([4], dtype=int64), array([5], dtype=int64), array([5], dtype=int64), array([5], dtype=int64), array([5], dtype=int64), array([5], dtype=int64
```

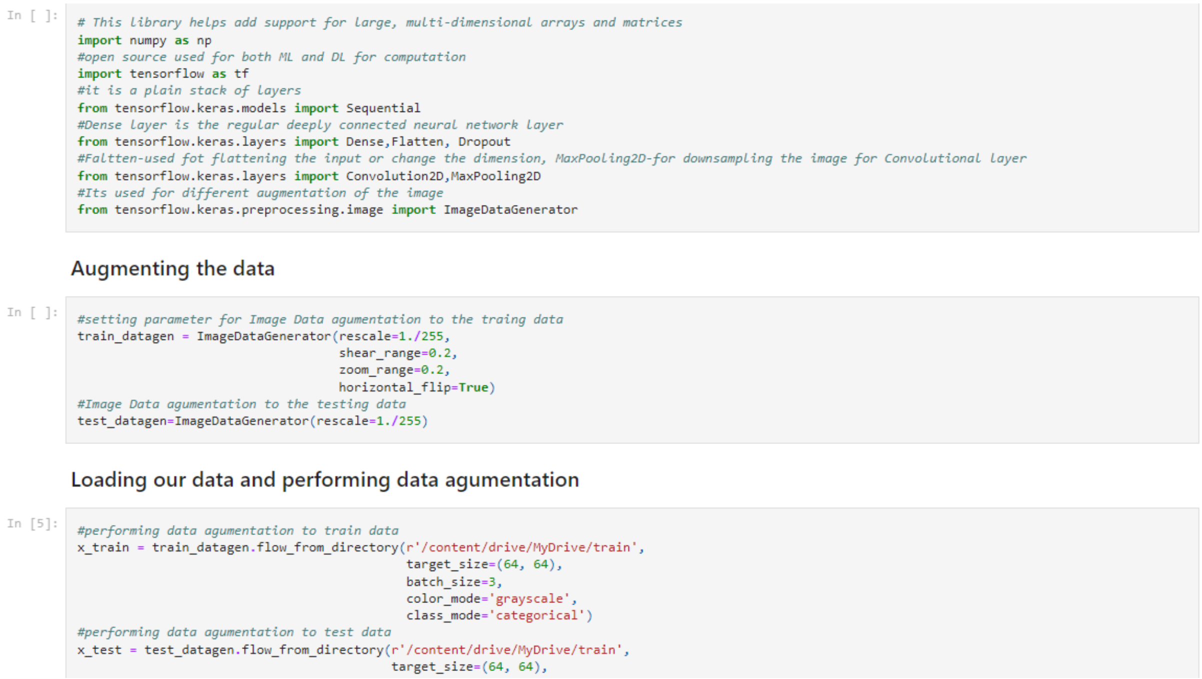
```
In [ ]: result = []
        index=['0','1','2','3','4','5']
        for i in p:
            result.append(index[i[0]])

        print(result)
```



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

        print(result)
```

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

MODEL BUILDING



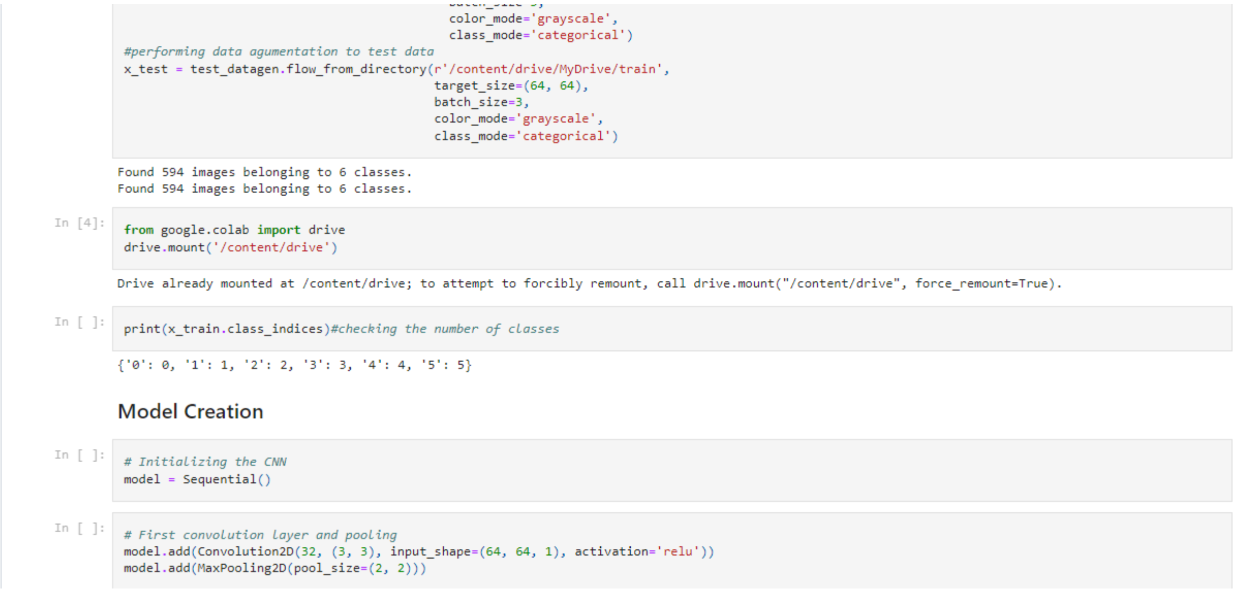
```
In [ ]:
# This library helps add support for large, multi-dimensional arrays and matrices
import numpy as np
#open source used for both ML and DL for computation
import tensorflow as tf
#it is a plain stack of layers
from tensorflow.keras.models import Sequential
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten, Dropout
#Faltten-used fot flattening the input or change the dimension, MaxPooling2D-for downsampling the image for Convolutional layer
from tensorflow.keras.layers import Convolution2D,MaxPooling2D
#Its used for different augmentation of the image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Augmenting the data

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

Loading our data and performing data agumentation

```
In [5]:
#performing data agumentation to train data
x_train = train_datagen.flow_from_directory(r'/content/drive/MyDrive/train',
                                            target_size=(64, 64),
                                            batch_size=3,
                                            color_mode='grayscale',
                                            class_mode='categorical')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(r'/content/drive/MyDrive/train',
                                          target_size=(64, 64),
```



```
                                            color_mode='grayscale',
                                            class_mode='categorical')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(r'/content/drive/MyDrive/train',
                                          target_size=(64, 64),
```

```
                                          batch_size=3,
                                          color_mode='grayscale',
                                          class_mode='categorical')
```

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

```
In [4]:
from google.colab import drive
drive.mount('/content/drive')
```

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

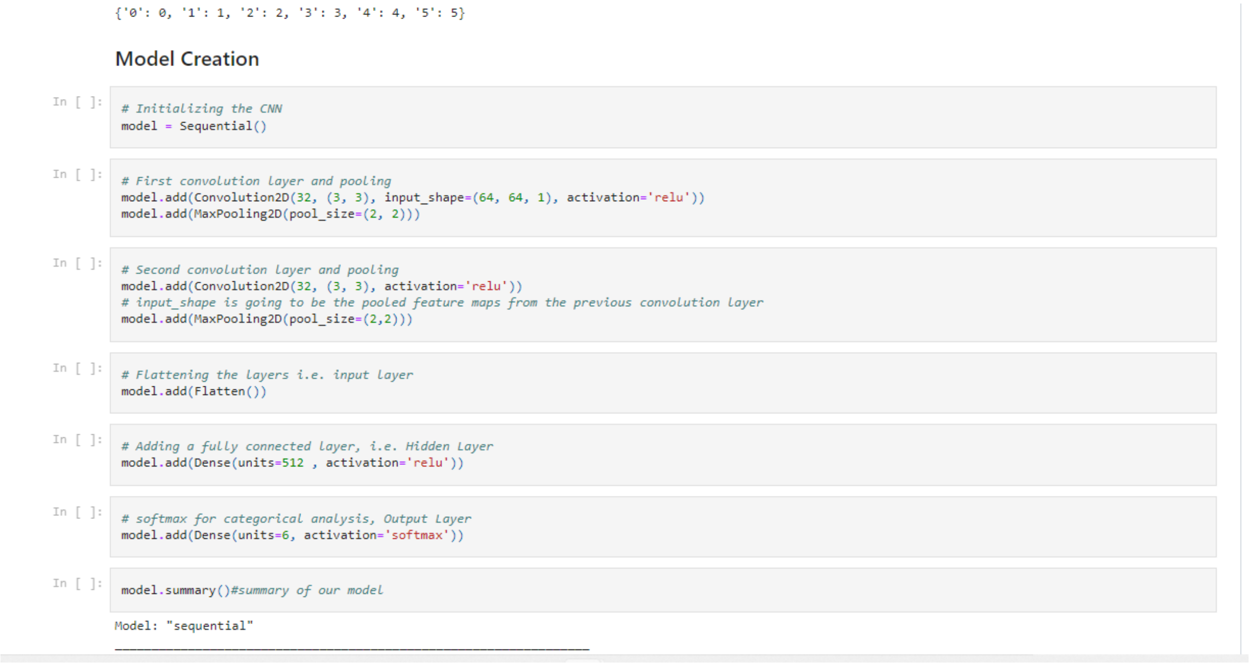
```
In [ ]:
print(x_train.class_indices)#checking the number of classes
```

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

Model Creation

```
In [ ]:
# Initializing the CNN
model = Sequential()
```

```
In [ ]:
# First convolution layer and pooling
model.add(Convolution2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```



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

Model Creation

```
In [ ]: # Initializing the CNN
        model = Sequential()
```

```
In [ ]: # First convolution layer and pooling
        model.add(Convolution2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
In [ ]: # Second convolution layer and pooling
        model.add(Convolution2D(32, (3, 3), activation='relu'))
        # input_shape is going to be the pooled feature maps from the previous convolution layer
        model.add(MaxPooling2D(pool_size=(2,2)))
```

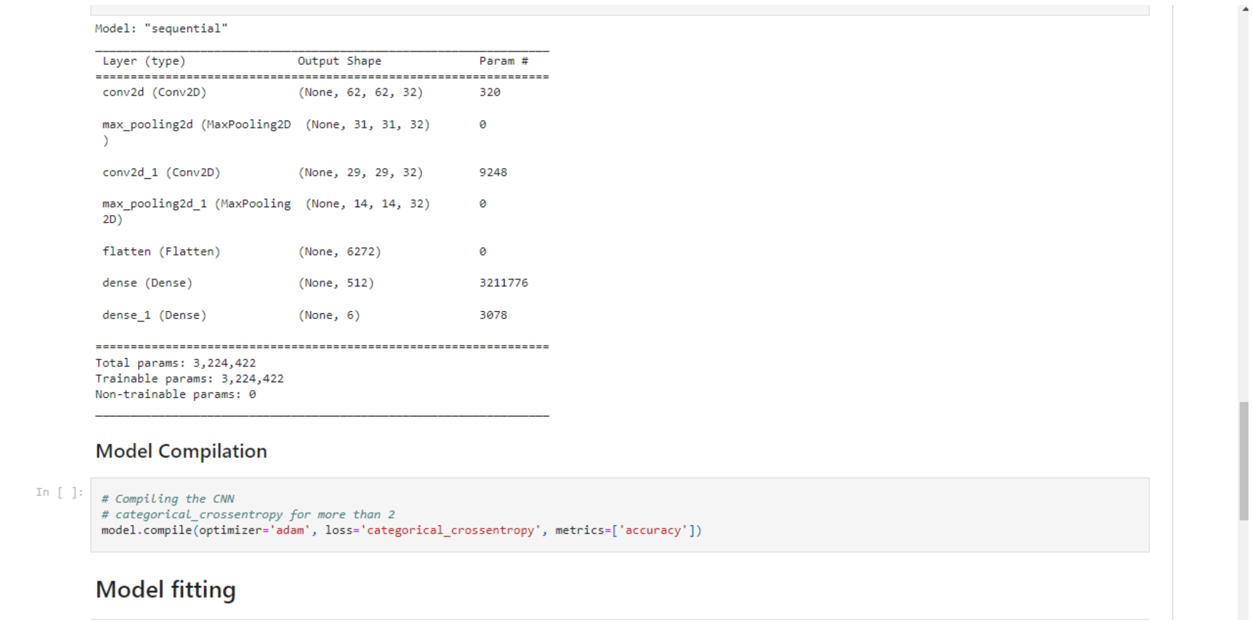
```
In [ ]: # Flattening the layers i.e. input layer
        model.add(Flatten())
```

```
In [ ]: # Adding a fully connected layer, i.e. Hidden Layer
        model.add(Dense(units=512 , activation='relu'))
```

```
In [ ]: # softmax for categorical analysis, Output Layer
        model.add(Dense(units=6, activation='softmax'))
```

```
In [ ]: model.summary()#summary of our model
```

```
Model: "sequential"
_________________________________________________________________
```

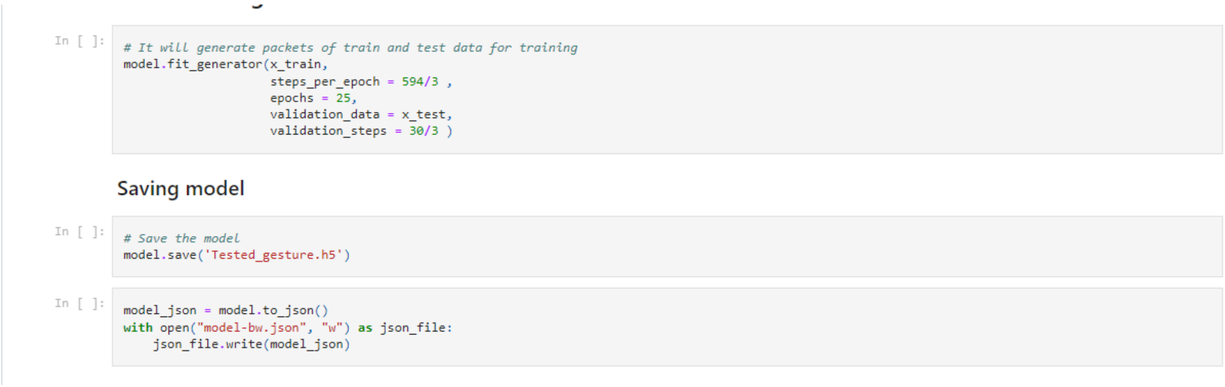


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        320

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

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

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

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 512)               3211776

 dense_1 (Dense)             (None, 6)                 3078

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

Model Compilation

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

Model fitting

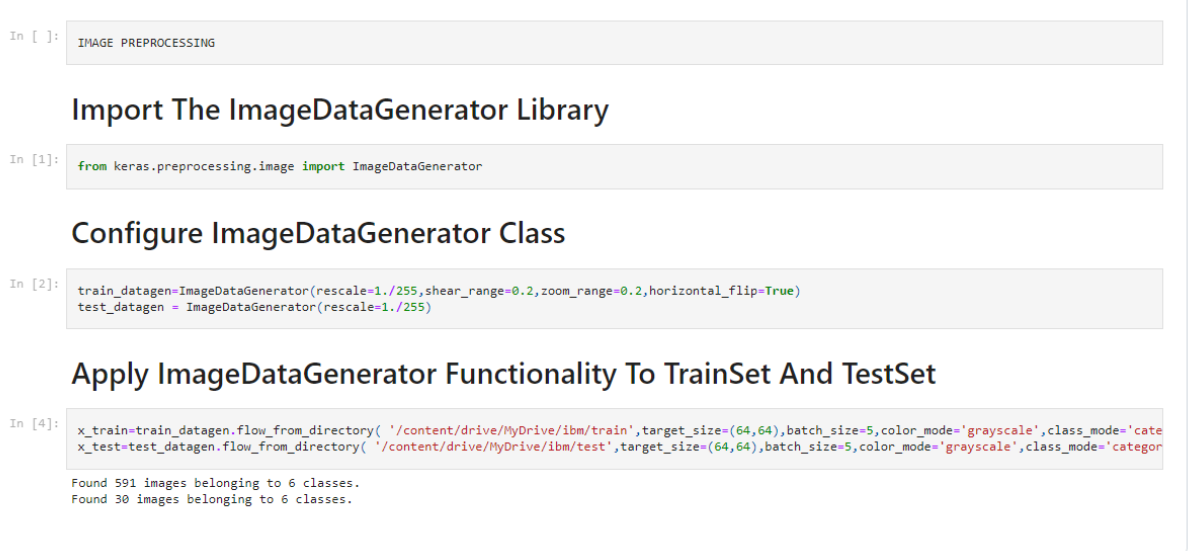


```
In [ ]: # It will generate packets of train and test data for training
        model.fit_generator(x_train,
                            steps_per_epoch = 594/3 ,
                            epochs = 25,
                            validation_data = x_test,
                            validation_steps = 30/3 )
```

Saving model

```
In [ ]: # Save the model
        model.save('Tested_gesture.h5')
```

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



In [ ]:

```
IMAGE PREPROCESSING
```

Import The ImageDataGenerator Library

In [1]:

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

Configure ImageDataGenerator Class

In [2]:

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

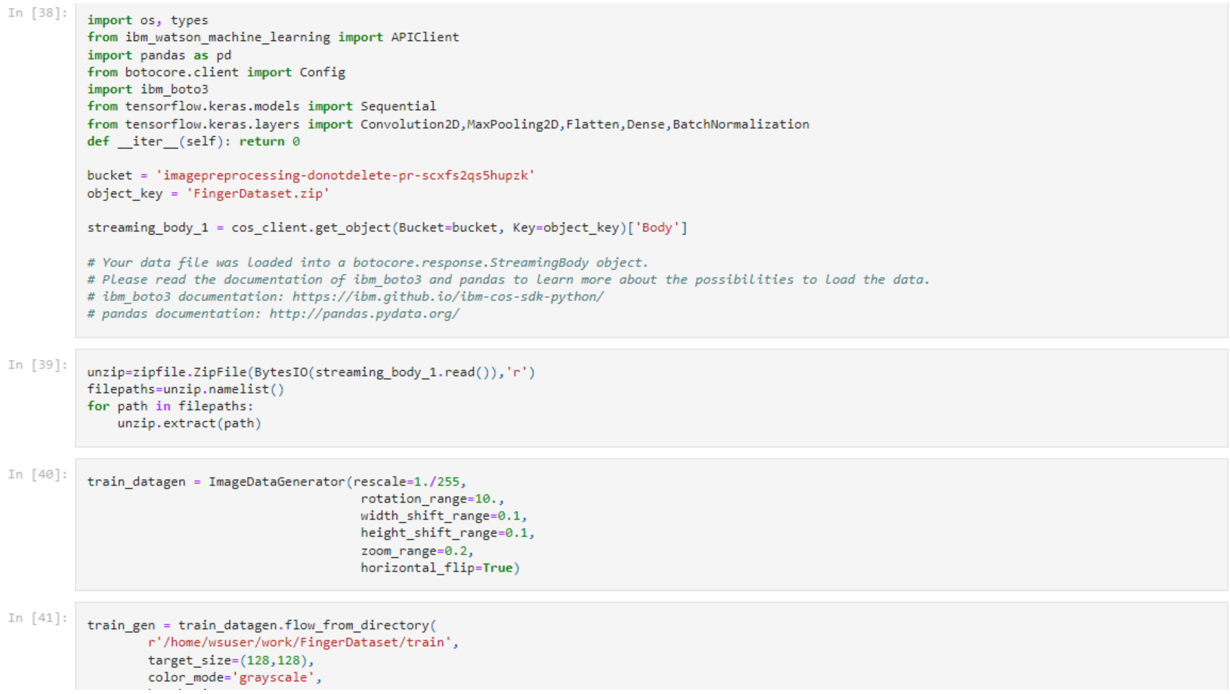
Apply ImageDataGenerator Functionality To TrainSet And TestSet

In [4]:

```
x_train=train_datagen.flow_from_directory( '/content/drive/MyDrive/ibm/train',target_size=(64,64),batch_size=5,color_mode='grayscale',class_mode='cate
x_test=test_datagen.flow_from_directory( '/content/drive/MyDrive/ibm/test',target_size=(64,64),batch_size=5,color_mode='grayscale',class_mode='categor
```

```
Found 591 images belonging to 6 classes.
Found 30 images belonging to 6 classes.
```

Deployment on IBM



In [38]:

```
import os, types
from ibm_watson_machine_learning import APIClient
import pandas as pd
from botocore.client import Config
import ibm_boto3
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense,BatchNormalization
def __iter__(self): return 0

bucket = 'imagepreprocessing-donotdelete-pr-scxfs2qs5hupzk'
object_key = 'FingerDataset.zip'

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

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [39]:

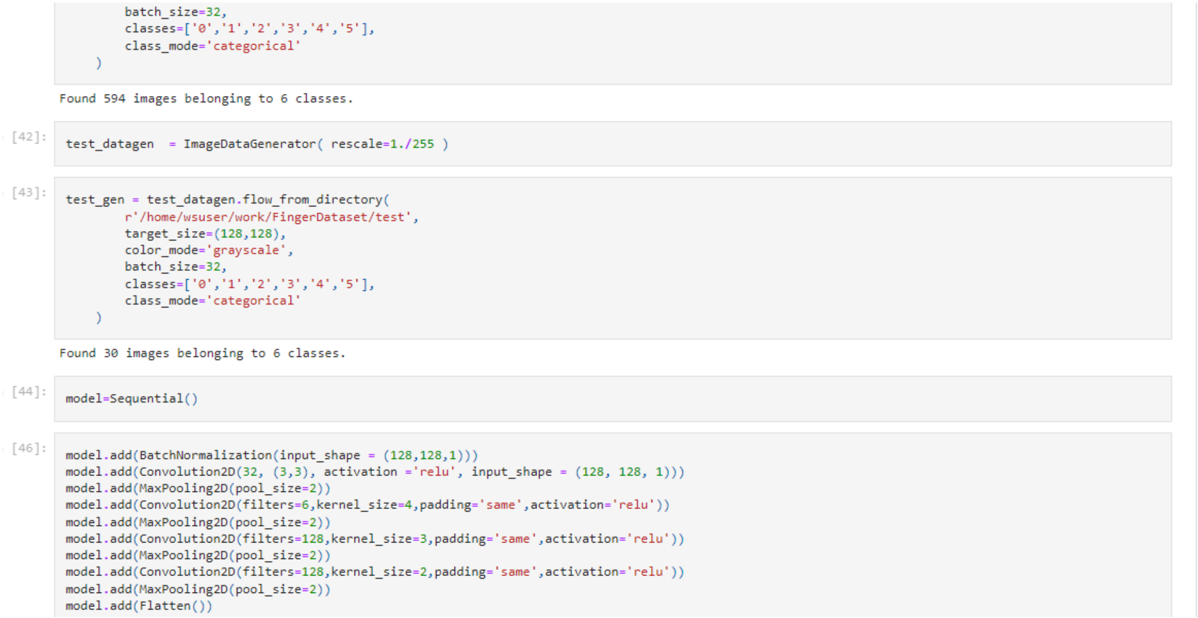
```
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
filepaths=unzip.namelist()
for path in filepaths:
    unzip.extract(path)
```

In [40]:

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                    rotation_range=10.,
                                    width_shift_range=0.1,
                                    height_shift_range=0.1,
                                    zoom_range=0.2,
                                    horizontal_flip=True)
```

In [41]:

```
train_gen = train_datagen.flow_from_directory(
        r'/home/wsuser/work/FingerDataset/train',
        target_size=(128,128),
        color_mode='grayscale',
```



```
        batch_size=32,
        classes=['0','1','2','3','4','5'],
        class_mode='categorical'
    )
```

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

[42]:

```
test_datagen  = ImageDataGenerator( rescale=1./255 )
```

[43]:

```
test_gen = test_datagen.flow_from_directory(
        r'/home/wsuser/work/FingerDataset/test',
        target_size=(128,128),
        color_mode='grayscale',
        batch_size=32,
        classes=['0','1','2','3','4','5'],
        class_mode='categorical'
    )
```

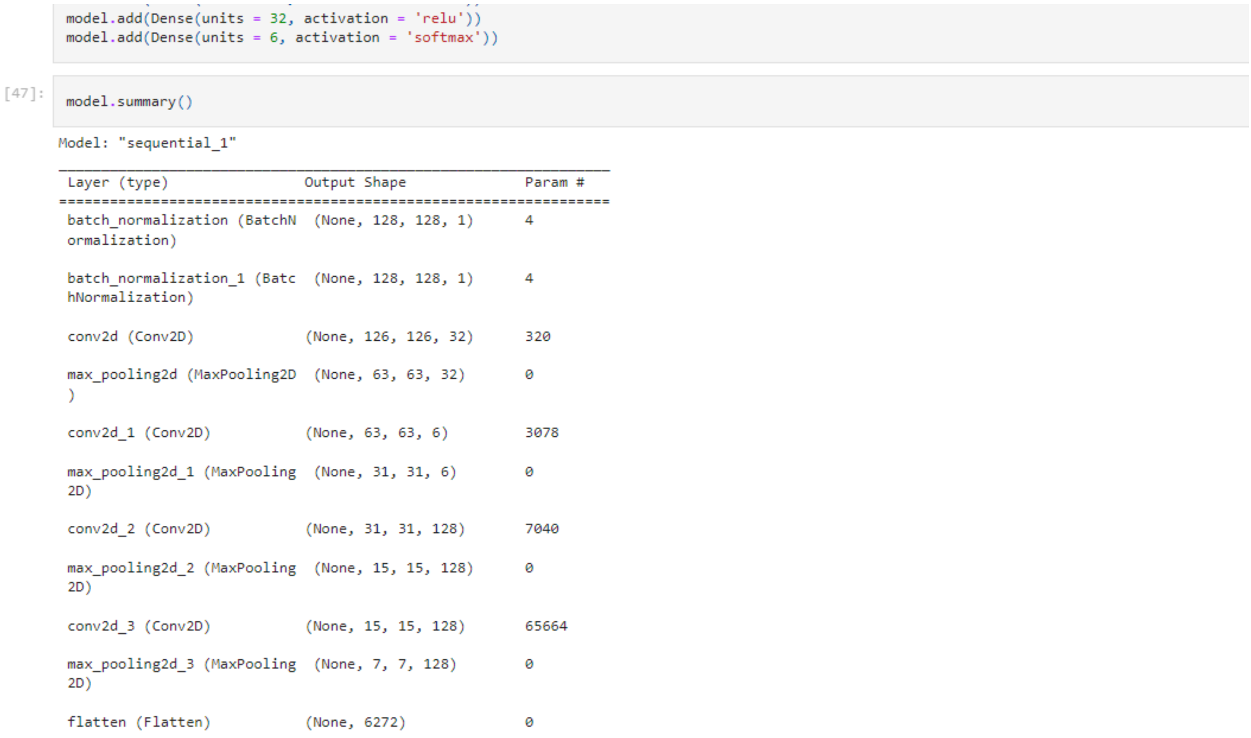
```
Found 30 images belonging to 6 classes.
```

[44]:

```
model=Sequential()
```

[46]:

```
model.add(BatchNormalization(input_shape = (128,128,1)))
model.add(Convolution2D(32, (3,3), activation ='relu', input_shape = (128, 128, 1)))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=6,kernel_size=4,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=3,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Convolution2D(filters=128,kernel_size=2,padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Flatten())
```

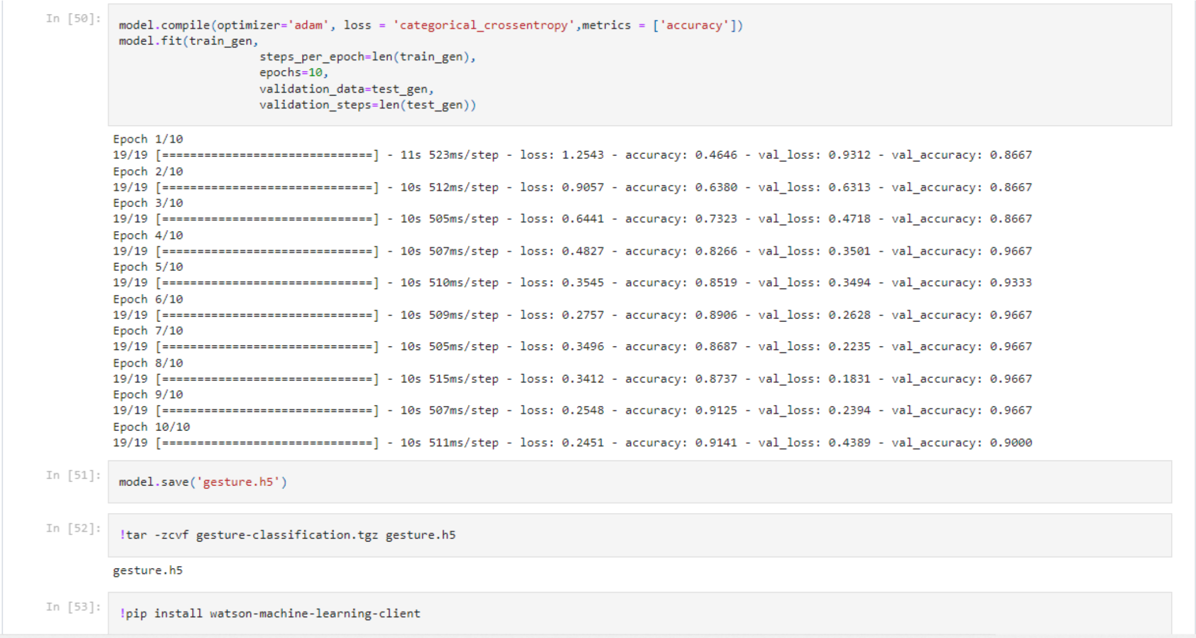


```
model.add(Dense(units = 32, activation = 'relu'))
model.add(Dense(units = 6, activation = 'softmax'))
```

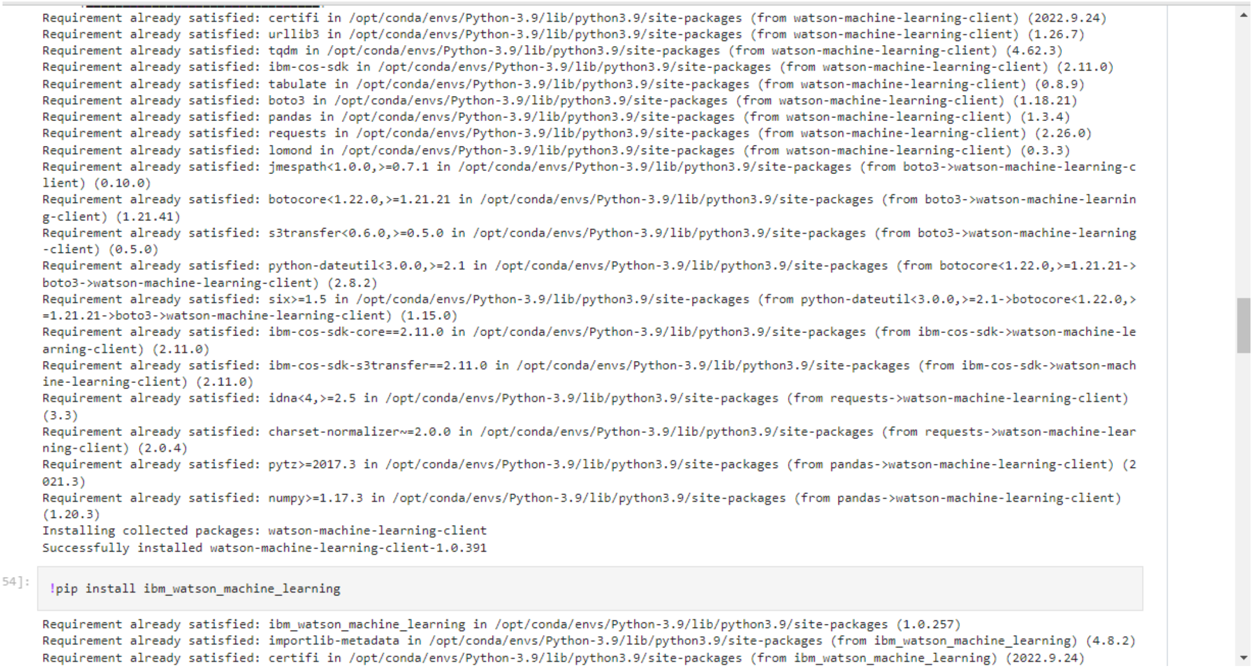
[47]:

```
model.summary()
```

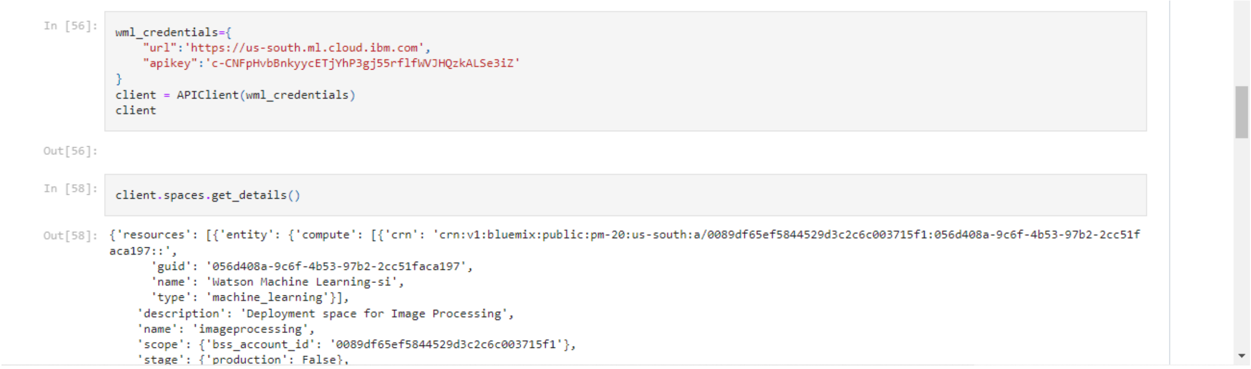
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 batch_normalization (BatchN  (None, 128, 128, 1)      4         
 ormalization)                                                   
                                                                 
 batch_normalization_1 (Batc  (None, 128, 128, 1)      4         
 hNormalization)                                                 
                                                                 
 conv2d (Conv2D)             (None, 126, 126, 32)      320       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0         
 )                                                               
                                                                 
 conv2d_1 (Conv2D)           (None, 63, 63, 6)         3078      
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 31, 31, 6)        0         
 2D)                                                             
                                                                 
 conv2d_2 (Conv2D)           (None, 31, 31, 128)       7040      
                                                                 
 max_pooling2d_2 (MaxPooling  (None, 15, 15, 128)      0         
 2D)                                                             
                                                                 
 conv2d_3 (Conv2D)           (None, 15, 15, 128)       65664     
                                                                 
 max_pooling2d_3 (MaxPooling  (None, 7, 7, 128)        0         
 2D)                                                             
                                                                 
 flatten (Flatten)           (None, 6272)              0         
```



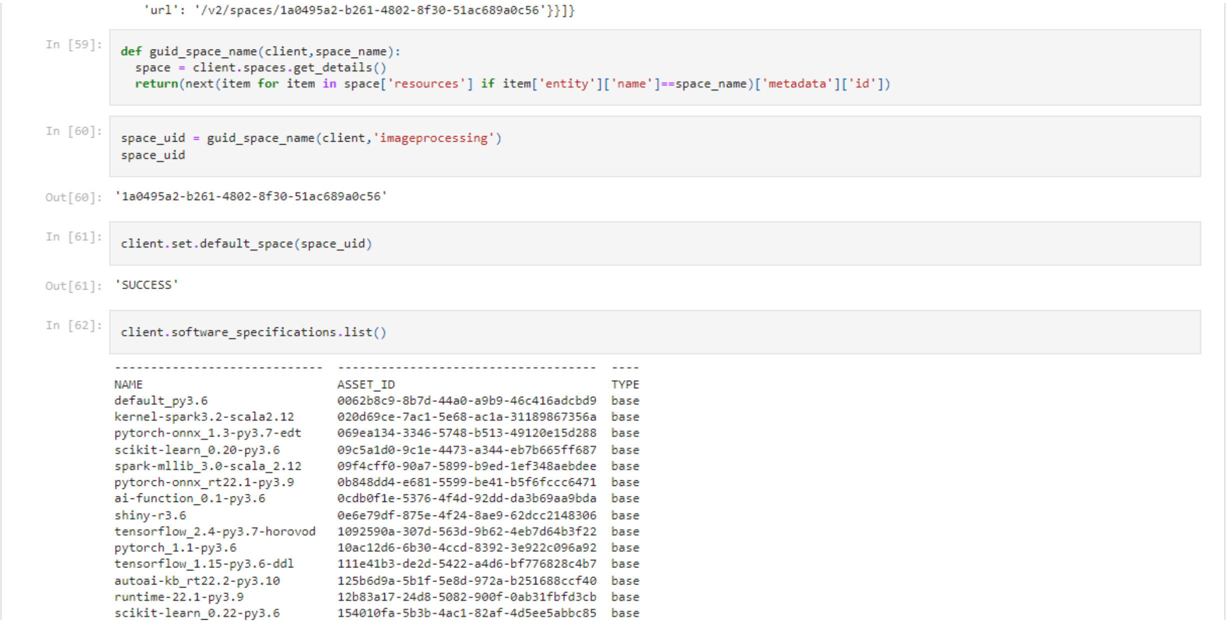
```
In [50]: model.compile(optimizer='adam', loss = 'categorical_crossentropy',metrics = ['accuracy'])
         model.fit(train_gen,
                             steps_per_epoch=len(train_gen),
                             epochs=10,
                             validation_data=test_gen,
                             validation_steps=len(test_gen))

         Epoch 1/10
         19/19 [==============================] - 11s 523ms/step - loss: 1.2543 - accuracy: 0.4646 - val_loss: 0.9312 - val_accuracy: 0.8667
         Epoch 2/10
         19/19 [==============================] - 10s 512ms/step - loss: 0.9057 - accuracy: 0.6380 - val_loss: 0.6313 - val_accuracy: 0.8667
         Epoch 3/10
         19/19 [==============================] - 10s 505ms/step - loss: 0.6441 - accuracy: 0.7323 - val_loss: 0.4718 - val_accuracy: 0.8667
         Epoch 4/10
         19/19 [==============================] - 10s 507ms/step - loss: 0.4827 - accuracy: 0.8266 - val_loss: 0.3501 - val_accuracy: 0.9667
         Epoch 5/10
         19/19 [==============================] - 10s 510ms/step - loss: 0.3545 - accuracy: 0.8519 - val_loss: 0.3494 - val_accuracy: 0.9333
         Epoch 6/10
         19/19 [==============================] - 10s 509ms/step - loss: 0.2757 - accuracy: 0.8906 - val_loss: 0.2628 - val_accuracy: 0.9667
         Epoch 7/10
         19/19 [==============================] - 10s 505ms/step - loss: 0.3496 - accuracy: 0.8687 - val_loss: 0.2235 - val_accuracy: 0.9667
         Epoch 8/10
         19/19 [==============================] - 10s 515ms/step - loss: 0.3412 - accuracy: 0.8737 - val_loss: 0.1831 - val_accuracy: 0.9667
         Epoch 9/10
         19/19 [==============================] - 10s 507ms/step - loss: 0.2548 - accuracy: 0.9125 - val_loss: 0.2394 - val_accuracy: 0.9667
         Epoch 10/10
         19/19 [==============================] - 10s 511ms/step - loss: 0.2451 - accuracy: 0.9141 - val_loss: 0.4389 - val_accuracy: 0.9000

In [51]: model.save('gesture.h5')

In [52]: !tar -zcvf gesture-classification.tgz gesture.h5

         gesture.h5

In [53]: !pip install watson-machine-learning-client
```



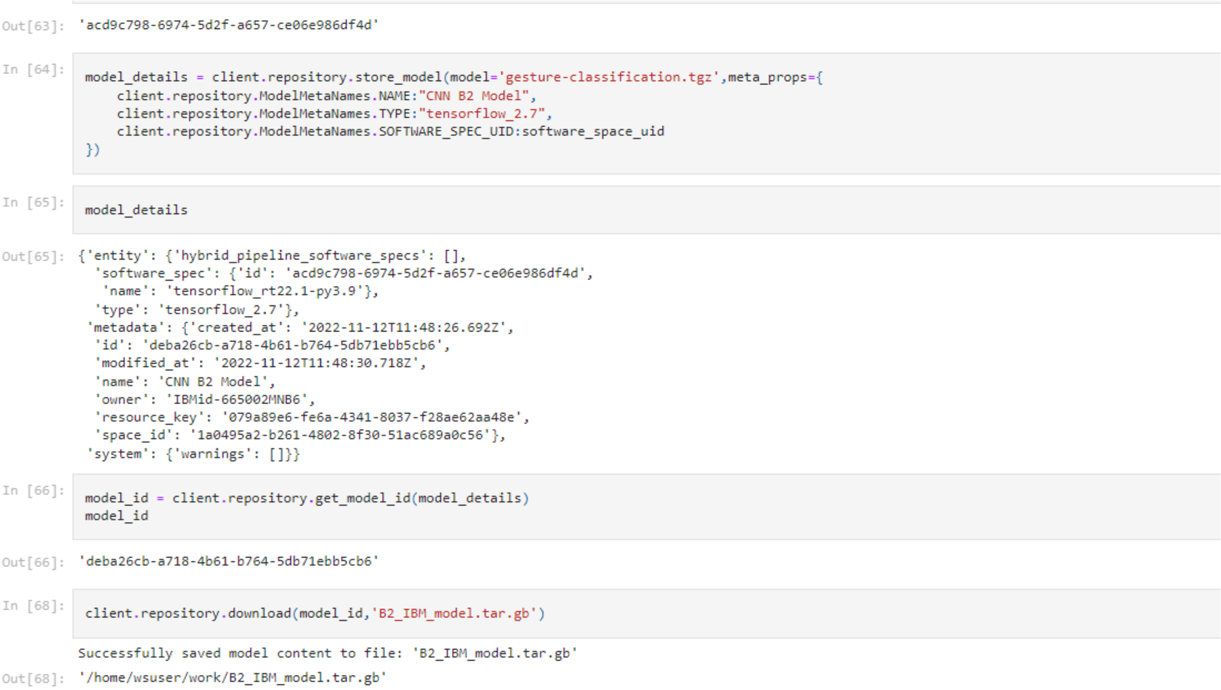
```
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-c
lient) (0.10.0)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learnin
g-client) (1.21.41)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning
-client) (0.5.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->
boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>
=1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-le
arning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-mach
ine-learning-client) (2.11.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client)
(3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-lear
ning-client) (2.0.4)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2
021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client)
(1.20.3)
Installing collected packages: watson-machine-learning-client
Successfully installed watson-machine-learning-client-1.0.391

54]: !pip install ibm_watson_machine_learning

Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (4.8.2)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2022.9.24)
```



```
In [56]: wml_credentials={
             "url":'https://us-south.ml.cloud.ibm.com',
             "apikey":'c-CNFpHvbBnkyycETjYhP3gj55rflfWVJHQzkALSe3iZ'
         }
         client = APIClient(wml_credentials)
         client

Out[56]:

In [58]: client.spaces.get_details()

Out[58]: {'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:us-south:a/0089df65ef5844529d3c2c6c003715f1:056d408a-9c6f-4b53-97b2-2cc51f
         aca197::',
                'guid': '056d408a-9c6f-4b53-97b2-2cc51faca197',
                'name': 'Watson Machine Learning-si',
                'type': 'machine_learning'}],
              'description': 'Deployment space for Image Processing',
              'name': 'imageprocessing',
              'scope': {'bss_account_id': '0089df65ef5844529d3c2c6c003715f1'},
              'stage': {'production': False},
```



```
           'url': '/v2/spaces/1a0495a2-b261-4802-8f30-51ac689a0c56'}}]}

In [59]: def guid_space_name(client,space_name):
           space = client.spaces.get_details()
           return(next(item for item in space['resources'] if item['entity']['name']==space_name)['metadata']['id'])

In [60]: space_uid = guid_space_name(client,'imageprocessing')
         space_uid

Out[60]: '1a0495a2-b261-4802-8f30-51ac689a0c56'

In [61]: client.set_default_space(space_uid)

Out[61]: 'SUCCESS'

In [62]: client.software_specifications.list()

         ----------------------------  ------------------------------------  ----
         NAME                          ASSET_ID                              TYPE
         default_py3.6                 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
         kernel-spark3.2-scala2.12     020d69ce-7ac1-5e68-ac1a-31189867356a  base
         pytorch-onnx_1.3-py3.7-edt    069ea134-3346-5748-b513-49120e15d288  base
         scikit-learn_0.20-py3.6       09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
         spark-mllib_3.0-scala_2.12    09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
         pytorch-onnx_rt22.1-py3.9     0b848dd4-e681-5599-be41-b5f6fccc6471  base
         ai-function_0.1-py3.6         0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
         shiny-r3.6                    0e6e79df-875e-4f24-8ae9-62dcc2148306  base
         tensorflow_2.4-py3.7-horovod  1092590a-307d-563d-9b62-4eb7d64b3f22  base
         pytorch_1.1-py3.6             10ac12d6-6b30-4ccd-8392-3e922c096a92  base
         tensorflow_1.15-py3.6-ddl     111e41b3-de2d-5422-a4d6-bf776828c4b7  base
         autoai-kb_rt22.2-py3.10       125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
         runtime-22.1-py3.9            12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
         scikit-learn_0.22-py3.6       154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
```



```
Out[63]: 'acd9c798-6974-5d2f-a657-ce06e986df4d'

In [64]: model_details = client.repository.store_model(model='gesture-classification.tgz',meta_props={
             client.repository.ModelMetaNames.NAME:"CNN B2 Model",
             client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
             client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
         })

In [65]: model_details

Out[65]: {'entity': {'hybrid_pipeline_software_specs': [],
           'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',
            'name': 'tensorflow_rt22.1-py3.9'},
           'type': 'tensorflow_2.7'},
          'metadata': {'created_at': '2022-11-12T11:48:26.692Z',
           'id': 'deba26cb-a718-4b61-b764-5db71ebb5cb6',
           'modified_at': '2022-11-12T11:48:30.718Z',
           'name': 'CNN B2 Model',
           'owner': 'IBMid-665002MNB6',
           'resource_key': '079a89e6-fe6a-4341-8037-f28ae62aa48e',
           'space_id': '1a0495a2-b261-4802-8f30-51ac689a0c56'},
          'system': {'warnings': []}}

In [66]: model_id = client.repository.get_model_id(model_details)
         model_id

Out[66]: 'deba26cb-a718-4b61-b764-5db71ebb5cb6'

In [68]: client.repository.download(model_id,'B2_IBM_model.tar.gb')

         Successfully saved model content to file: 'B2_IBM_model.tar.gb'
Out[68]: '/home/wsuser/work/B2_IBM_model.tar.gb'
```

Github Link

https://github.com/IBM-EPBL/IBM-Project-15052-1659593576

Project Demo Link

https://drive.google.com/file/d/1pTX9phZ3Mv6pKZ2sSLj9fcDvaNgW0H17/view?usp=share_link