

A GESTURE-BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

A project report submitted in partial fulfillment of the
requirements of the award of the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

By

P. N. SESA SAI SARMA (410119106044)

B. NAVEEN SAI (410119106007)

Md. DANISH FURKAN (410119106035)

M. RAMYA (410119106037)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

ADHI COLLEGE OF ENGINEERING AND TECHNOLOGY

(Accredited by NAAC, Affiliated by ANNA UNIVERSITY, Chennai & Approved by AICTE)

KANCHIPURAM (Dt.), Chennai - 631605

2022

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING

ADHI COLLEGE OF ENGINEERING AND TECHNOLOGY



CERTIFICATE

This is to certify that the project report titled “A Gesture Based Tool For Sterile Browsing Of Radiology Images”, being submitted by

P. N. SESA SAI SARMA (410119106044)

B. NAVEEN SAI (410119106007)

Md. DANISH FURKAN (410119106035)

M. RAMYA (410119106037)

in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics And Communication Engineering, to the Anna University, Chennai is a record of bonafied work carried out by them my guidance and supervision.

Faculty Mentor

ANNIE ANGELINE PREETHI S

Industry Mentor

MS. PRADEEPTHI

CONTENTS

- 1. INTRODUCTION**
 - Project Overview
 - Purpose
- 2. LITERATURE SURVEY**
 - Existing problem
 - References
 - Problem Statement Definition
- 3. IDEATION & PROPOSED SOLUTION**
 - Empathy Map Canvas
 - Ideation & Brainstorming
 - Proposed Solution
 - Problem Solution fit
- 4. REQUIREMENT ANALYSIS**
 - Functional requirement
 - Non-Functional requirements
- 5. PROJECT DESIGN**
 - Data Flow Diagrams
 - Solution & Technical Architecture
 - User Stories
- 6. PROJECT PLANNING & SCHEDULING**
 - Sprint Planning & Estimation
 - Sprint Delivery Schedule
- 7. CODING & SOLUTIONING**
 - Feature 1
 - Feature 2
- 8. TESTING**
 - Test Cases
 - User Acceptance Testing
- 9. RESULTS**
- 10. ADVANTAGES & DISADVANTAGES**
- 11. CONCLUSION**
- 12. FUTURE SCOPE**
- 13. APPENDIX**
 - Source Code
 - GitHub & Project Demo Link

ABSTRACT

Touch less hand gesture recognition systems are becoming important in automotive user interfaces as they. Improve safety and comfort. Various computer vision algorithms have employed color and depth cameras for hand gesture recognition, but robust classification of gestures from different subjects performed under widely varying lighting conditions is still challenging. We propose an algorithm for drivers' hand gesture recognition from challenging depth and intensity data using 3D convolutional neural networks. Our solution combines information from multiple spatial scales for the final prediction. It also employs spatio-temporal data augmentation form ore effective training and to reduce potential over fitting.

1.INTRODUCTION

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 technique 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 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 grayscale etc.

Purpose:

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

2. LITERATURE SURVEY

Existing problem:

Hand Gesture Recognition Using Camera ” is based on concept of Image processing. In recent year there is lot of research on gesture recognition using kinect sensor on using HD camera but camera and kinect sensors are more costly Hand gestures come naturally to us. It is even found that blind people use hand gestures where speaking with other blind people. Gesturing helps you access memories. Using hand gestures while you speak not only helps others remember what you say it also helps you speak more quickly and effectively!

References:

The fundamental commitments and extent of investigation can be characterized as follows:

We have examined different accessible radar innovations to understand their similitudes and contrasts. Every one of the viewpoints connected with HGR acknowledgment, including information obtaining, information portrayal, information preprocessing and grouping, are made sense of exhaustively.

We made sense of the radar-recorded hand-signal information portrayal methods for 1D, 2D and 3D classifiers. In view of this information portrayal, subtleties of the accessible HGR calculations are talked about.

At long last, subtleties connected with application-arranged HGR research works are likewise introduced. A few patterns and study examinations are like wise include

3. IDEATION & PROPOSED SOLUTION

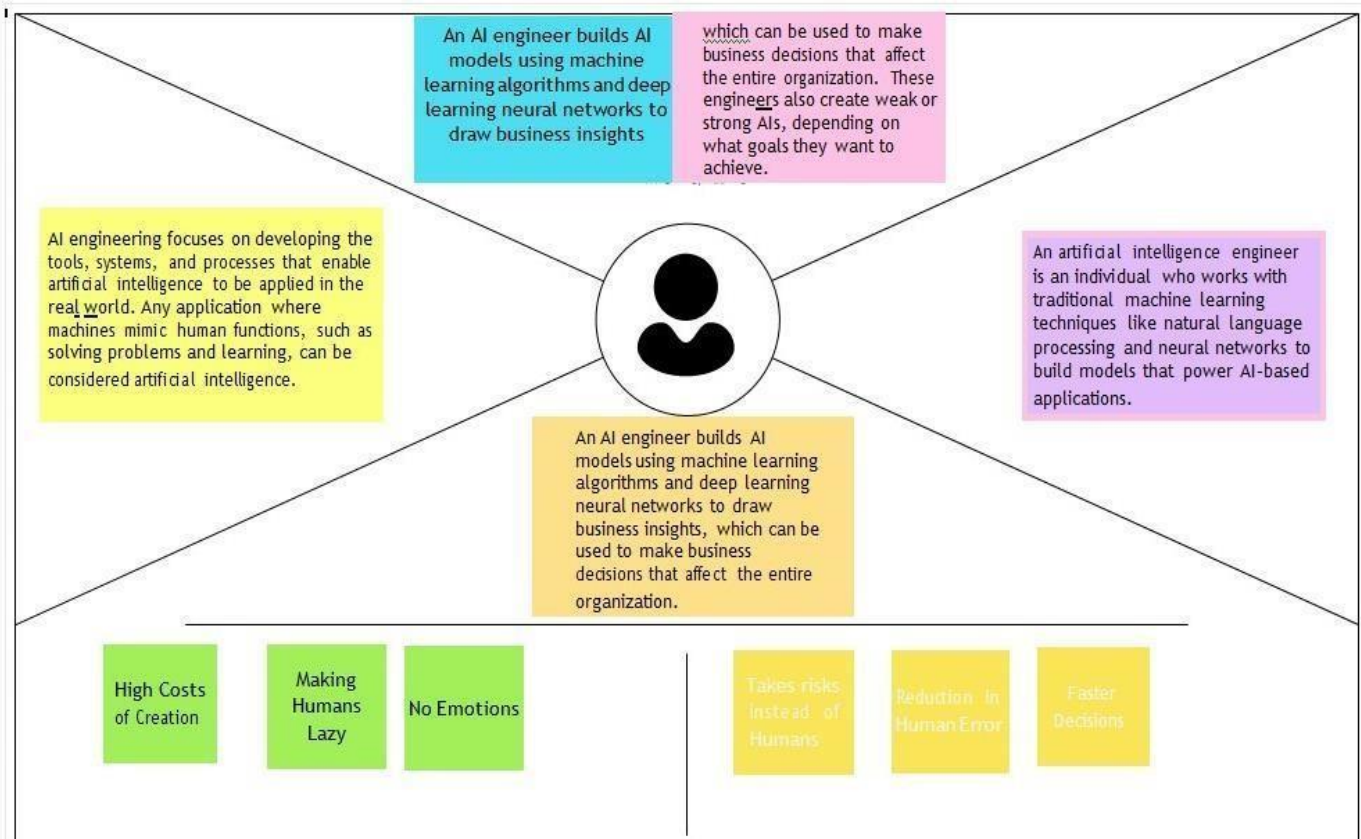
3.1 Empathy Map Canvas:

Empathy Map Canvas

Gain insight and understanding on solving customer problems.



Build empathy and keep your focus on the user by putting yourself in their shoes.



Share your feedback

3.1 Ideation & Brainstorming

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions to generate ideas, clarify their inspiration and categorize ideas to help you find something to take on.

1. Set up your session
2. Set up your session
3. Set up your session

Define your problem statement

What problem are you trying to solve? Frame your problem in a way that is clear and specific. The problem should be a question.

1. Problem statement

In this project, Gesture-based Desktop automation, i.e. the model is trained on the images of different hand gestures, such as showing numbers with fingers etc. The 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. The gesture predicts the then images to be shown. Image is rotated 3 times, it is rotated etc.

Brainstorm

What ideas do you have for solving the problem? Write down as many ideas as you can. Do not worry about whether they are good or bad. The goal is to generate as many ideas as possible.

1. Brainstorm

SESHA SAI SHERJIA

- Hand gesture
- Image processing
- Machine learning

SENAKALA NAVEEN

- Hand gesture
- Image processing
- Machine learning

DANISH FURKAN

- Hand gesture
- Image processing
- Machine learning

RAHITA

- Hand gesture
- Image processing
- Machine learning

Group ideas

Sort the ideas into groups with similar themes or related concepts. You can do this by putting the ideas into groups based on their similarity. You can also group ideas by their complexity, by their feasibility, or by their impact.

1. Group ideas

WE NEED TO COLLECT THE DATA

DEVELOPING GESTURES THROUGH SIGNAL PROCESSING

WE SHOULD USE DETECT THROUGH SENSORS

WE NEED TO ADD THE DATA ABOUT FINGER COUNTS

BY USING MULTIPLE FINGER DETECTION WE CAN GET MULTIPLE OUTPUT

WITHIN IN FRACTION OF SECONDS WE CAN GET OUTPUT

BY USING ACTIONS WE CAN GET THE OUTPUT

BY USING ALGORITHMS WE CAN GET STEP BY STEP PROCESS TO GET OUTPUT

Prioritize

Sort the ideas into groups based on their feasibility and impact. You can do this by putting the ideas into groups based on their similarity. You can also group ideas by their complexity, by their feasibility, or by their impact.

1. Prioritize

Proposed Solution Template:

Project team shall fill the following information in proposed solution templates.

S. No	Parameter	Description
1.	Problem Statement (problem to be solved)	"Hand Gesture Recognition Using Camera " is based on concept of Image processing. In recent year there is lot of research on gesture recognition using kinect sensor on using HD camera but camera and kinect sensors are more costly.
2.	Idea / Solution description	Hand gestures come naturally to us. It is even found that blind people use hand gestures when speaking with other blind people. Gesturing helps you access memories. Using hand gestures while you speak not only helps others remember what you say, it also helps you speak more quickly and effectively!
3.	Novelty / Uniqueness	In this proposed system, "Kinect camera" plays the major role to gather the depth information from the skeleton.
4.	Social Impact / Customer Satisfaction	These results suggest that gesture aids the listener as well as the speaker and that gesture has a direct effect on listener

3.4 Problem Solution Fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) ●This tools was designed for doctor for support medical imaging manipulation. ●Doctors(22+ ages)	6. CUSTOMER CONSTRAINTS ●Data Privacy ●Technology awareness ●Customer should have uninterrupted connection.	5. AVAILABLE SOLUTIONS ●In early stage the doctors has to manually do navigation and manipulation of images in an electronic medical record(EMR). ●By using the gesture-based navigation and manipulation of images is very much useful for the doctors.
---	--	--

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS ●Reduce time ●Produce more accurate solution for the report(EMR). ●Easy navigation and manipulation of images.	9. PROBLEM ROOT CAUSE ●Doctor can't able to see each and every patient records. ●User friendly and doctor friendly services, ●These technologies are expensive right mom	7. BEHAVIOUR ●If any problem or technical issue in software faced by our customer, they will send us feedback on the same and our technical team will solve their problem in efficient way and get back to them by sending mail.
---	---	---

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

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Data collection	Collection of required input data
FR-4	Data analysis	Process the given inputs using CNN and Nutrion API
FR-5	Data processing	Evaluate the data and store it in database and integrate in cloud containers
FR-6	Provide output to user	Display the result to the user

4.2 Non-Functional requirements

Non-functional Requirements:

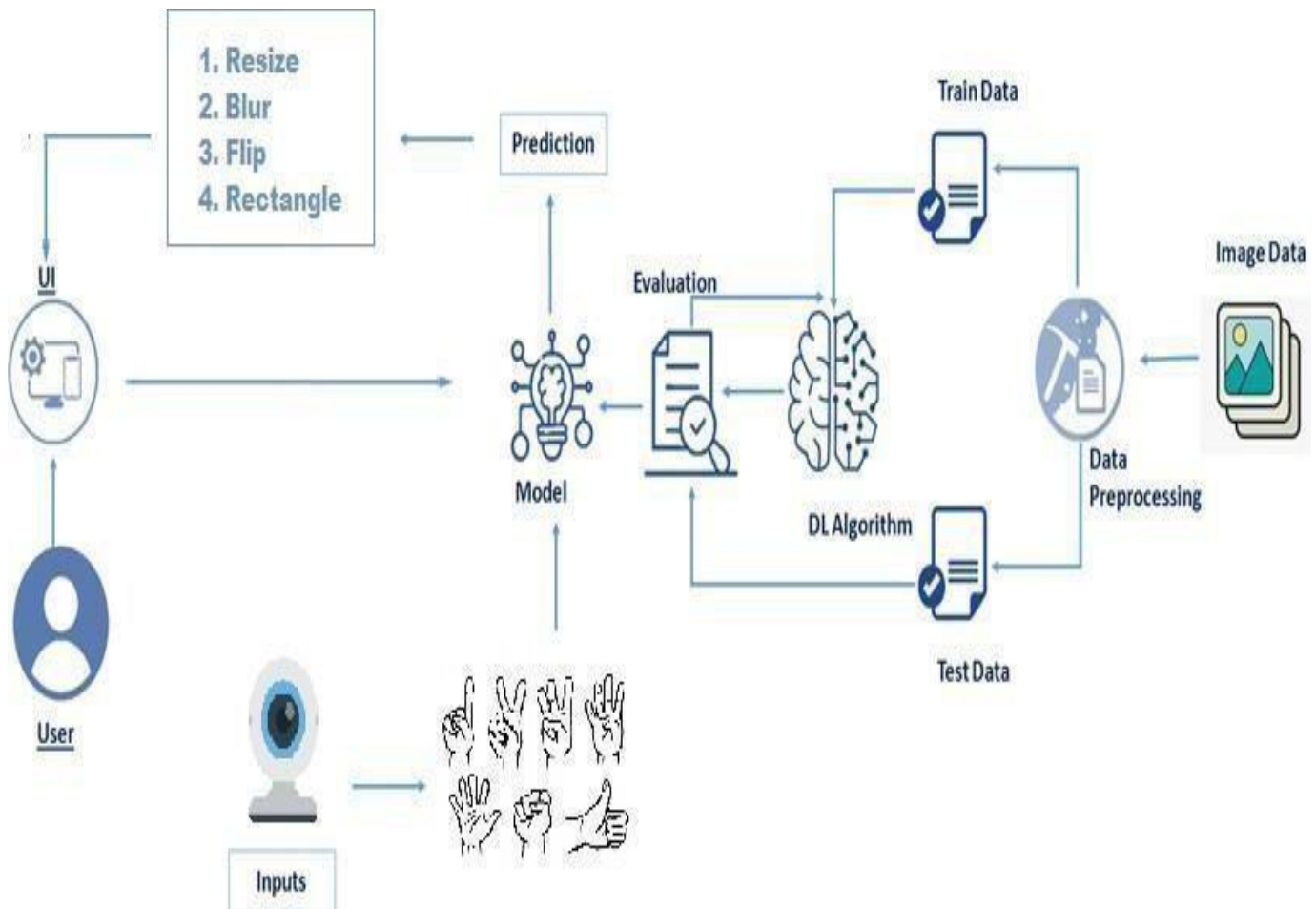
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	User-friendly and overall satisfaction of the user while using the website
NFR-2	Security	The website provides proper authentication and verification
NFR-3	Reliability	The site always provides reliable outputs and lacks failures
NFR-4	Performance	Provides 100% efficiency of the output
NFR-5	Availability	The product is readily available for all kinds of users when needed
NFR-6	Scalability	Effective in obtaining good accuracies

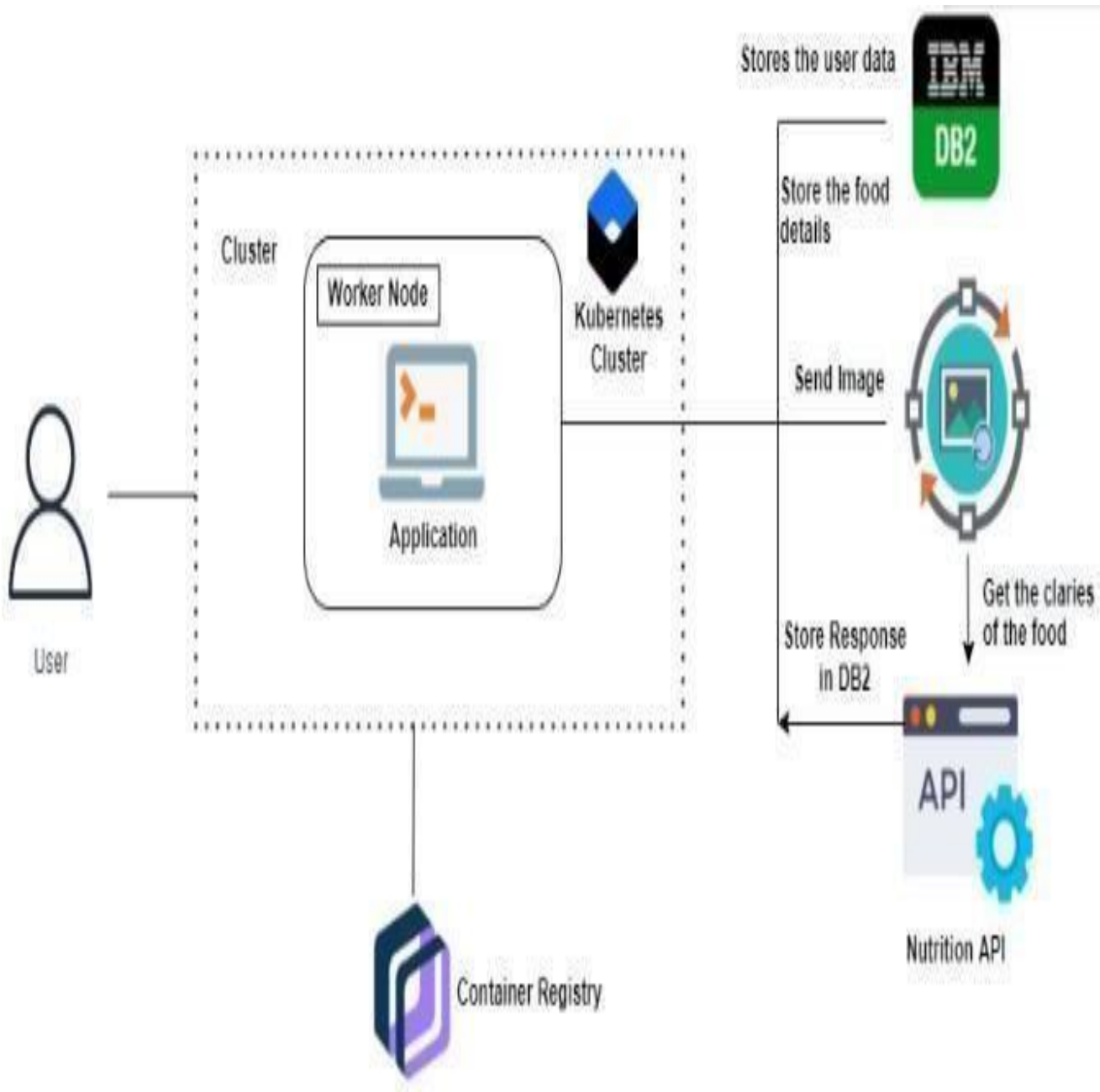
5.

PROJECT DESIGN

Solution & Technical Architecture



Technical Architecture



6. PROJECT PLANNING & SCHEDULING:

Sprint Planning & Estimation:

1. Defining our classification categories
2. Collect training images
3. Train the model
4. Test our model

User interacts with the UI (User Interface) to upload the image as input. Depending on the different gesture inputs, different operations are applied to the input image.

Once the model analyses the gesture, the prediction with operation applied on the image is how it is cased on the UI.

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

listed below: Data Collection.

Collect the dataset or Create the dataset

Data Pre processing

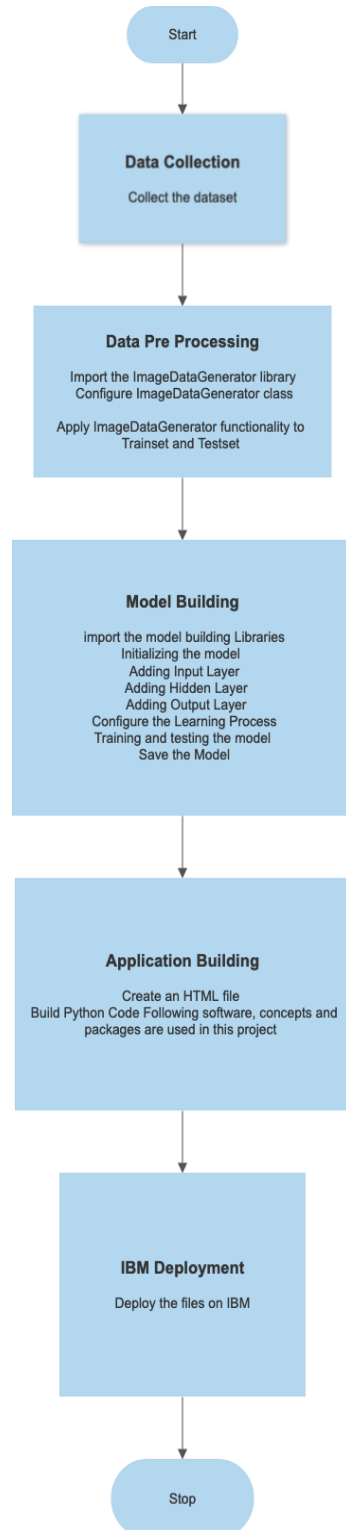
Import the ImageDataGenerator

library

Configure ImageDataGenerator class

- Apply ImageDataGenerator functionality to Trainset and Testset
- Model Building
 - Import the model building Libraries
 - Initializing the model
 - Adding Input Layer
 - Adding Hidden Layer
 - Adding Output Layer
 - Configure the Learning Process
 - Training and testing the model
 - Save the Model
- Application Building
 - Create an HTML file
 - Build Python Code

Sprint Delivery Schedule:



7. CODING & SOLUTIONING

Feature-1

Importing the Model Building Libraries

```
In [1]: import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers

from tensorflow.keras.layers import Dense, Flatten

from tensorflow.keras.layers import Conv2D, MaxPooling2D
from keras.preprocessing.image import ImageDataGenerator

Using TensorFlow backend.
```


Initializing the Model

```
In [9]: # Initializing the CNN  
classifier = Sequential()
```



Sequential model is a linear stack of layers.



You can create a sequential model by passing a list of layer instances to the constructor from `keras.models` import `Sequential` from `keras`

Adding CNN Layers

```
# First convolution layer and pooling  
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu'))  
classifier.add(MaxPooling2D(pool_size=(2, 2)))  
# Second convolution layer and pooling  
classifier.add(Conv2D(32, (3, 3), activation='relu'))  
# input_shape is going to be the pooled feature maps from the previous convolution layer  
classifier.add(MaxPooling2D(pool_size=(2, 2)))  
  
# Flattening the layers  
classifier.add(Flatten())
```

We are adding a convolution layer with activation function as “`relu`” and with a small filter size (3,3) and number of filters (32) followed by a max pooling layer



Maxpool layer is used to downsample the input



Flatten layer flattens the output

Adding Dense Layers

- Dense layer is deeply connected neural network layer. It is most common and frequently used layer.
- 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 models and its layers.

```
In [10]: # Adding a fully connected layer
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=6, activation='softmax')) # softmax for more than 2
```

```
classifier.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape
=====	
conv2d (Conv2D)	(None, 62, 62, 32)

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

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

max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)

flatten (Flatten)	(None, 6272)

dense (Dense)	(None, 128)

dense_1 (Dense)	(None, 6)
=====	
Total params: 813,286	
Trainable params: 813,286	
Non-trainable params: 0	

8. TESTING

Test Cases

Train the model:



Fit_generator functions are used to train a deep learning neural network



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.

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

Validation data can be either

- An inputs and targets list
- A generator
- 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.

```
In [13]: classifier.fit_generator(  
        generator=x_train, steps_per_epoch = len(x_train),  
        epochs=50, validation_data=x_test, validation_steps = len(x_test))
```

Save the model:

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

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

- The model is saved with .h5 extension.
- An H5 file is a data file saved in the Hierarchical Data Format (HDF)
- It contains multidimensional arrays of scientific data

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

```
In [24]: index=['0','1','2','3','4','5']
result=str(index[pred[0]])
result
```

```
Out[24]: '3'
```

```
In [23]: img = image.load_img(r"C:\Manisha\3.jpg", grayscale=True, target_size= (64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred = model.predict_classes(x)
pred
```

```
Out[23]: array([3], dtype=int64)
```

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

User Acceptance

Testing: Configure the

Learning Process

Configure the Learning Process



The compilation is the final step in creating the model. Once the compilation is done, we can move on to the training process.



Optimization is an important process which optimize the input weights by comparing the prediction and the loss function



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

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

Source Code:

Build Python Code

Build flask file 'app.py' which is a web framework written in python for server-side scripting.

App starts running when “_name_” constructor is called in main.

render_template is used to return html file.

“GET” method is used to take input from the user.

“POST” method is used to display the output to the user.

Importing Libraries

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

from tensorflow.keras.models import load_model # to load our trained model
import os
from werkzeug.utils import secure_filename
```

- Creating our flask application and loading our model:

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

- Routing to the html Page

```
@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("index6.html")
```

The above three routes are used to render the home, introduction and the indexhtml pages.

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

And the predict route is used for prediction and it contains all the codes which are used for predicting our results. Firstly, inside launch function we are having the following things:

- Getting our input and storing it
- Grab the frames from the web cam.
- 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 predict 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 so function we will be storing the image in the uploads folder in our local system

Getting our input and storing it

```
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)
```

Grab the frames from the web cam

```
cap = cv2.VideoCapture(0)
while True:
    _, frame = cap.read() #capturing the video frame values
    # Simulating mirror image
    frame = cv2.flip(frame, 1)
```

Creating ROI

```
# Got this from collect-data.py
# Coordinates of the ROI
x1 = int(0.5*frame.shape[1])
y1 = 10
x2 = frame.shape[1]-10
y2 = int(0.5*frame.shape[1])
# Drawing the ROI
# The increment/decrement by 1 is to compensate for the bounding box
cv2.rectangle(frame, (x1-1, y1-1), (x2+1, y2+1), (255,0,0), 1)
# Extracting the ROI
roi = frame[y1:y2, x1:x2]

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

Predicting our results

```
result = model.predict(test_image.reshape(1, 64, 64, 1))
prediction = {'ZERO': result[0][0],
             'ONE': result[0][1],
             'TWO': result[0][2],
             'THREE': result[0][3],
             'FOUR': result[0][4],
             'FIVE': result[0][5]}

# Sorting based on top prediction
prediction = sorted(prediction.items(), key=operator.itemgetter(1), reverse=True)

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



```

#loading an image
image1=cv2.imread(file_path)
if prediction[0][0]=='ONE':

    resized = cv2.resize(image1, (200, 200))
    cv2.imshow("Fixed Resizing", resized)
    key=cv2.waitKey(3000)

    if (key & 0xFF) == ord("1"):
        cv2.destroyAllWindows("Fixed Resizing")

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

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

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

elif prediction[0][0]=='THREE':
    blurred = cv2.GaussianBlur(image1, (11, 11), 0)
    cv2.imshow("Blurred", blurred)
    key=cv2.waitKey(3000)
    if (key & 0xFF) == ord("3"):
        cv2.destroyAllWindows("Blurred")
else:
    continue

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

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

```


9. RESULT

Final findings (Output) of the project along with screen shots. Through this project we found that we can maintain the sterility of an operation theater, etc by using hand based gesture tools to browse the images obtained.

10. ADVANTAGES & DISADVANTAGES

Advantages:

Major advantage of this tool is that it helps to maintain the sterility of the environment. It is also easy to use and is quicker than the existing methods to browse images.

It can also be performed even if the surgeon is a bit far away from the system, this helps to save time.

The tool does not need the person using it to have an apparatus or any devices on them to use it. They can simply move their hands to browse through the images.

Disadvantages:

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

APPLICATIONS

This hand based gesture tool developed can be mainly used in the medical industry to browse images without compromising the sterility.

However it can also be used in different industries while presenting certain ideas, during meetings, and can be used by teachers while teaching.

11. CONCLUSION

In this project we developed a tool which recognizes 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's industries like it can be used by present by teachers for show images in the classroom.

FUTURE SCOPE

- The tools can be made quicker by increasing the recognition speed.
- More number of gestures can be added there by increasing this tool's functionalities and abilities 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.

REFERENCES

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2410001/>
<https://pubmed.ncbi.nlm.nih.gov/18451034/>
https://www.researchgate.net/publication/5401674_A_Gesture
https://www.researchgate.net/publication/5401674_A_Gesture

APPENDIX

Git hub link:

<https://github.com/IBM-EPBL/IBM-Project-14651-1659588154>

Video link:

<https://www.awesomescreenshot.com/video/12624629?key=d7e1691e590a1a1c1c48c83f7021b357>