Mini Project Report

On

COMPUTER NAVIGATION WITH HAND GESTURE RECOGNITION

Submitted by

Uzma Ali (Roll.No.- 67)

Manish Walurkar (Roll.No.- 71)

Rohit Farkade (Roll.No.- 14)

Mohammad Gaznavi Sheikh (Roll.No.- 59)

Mansi Bambode (Roll.No.- 03)

Guided by

Prof. C. U. Chauhan

Prof. S. W. Shende

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING

(An institution affiliated to Gondwana University Gadchiroli)

CHANDRAPUR – 442 403

2023-2024

GOVERNMENT COLLEGE OF ENGINEERING, CHANDRAPUR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Session: 2023-24

Certificate

This is to certify that

Ms. Uzma Jabeen Syed Raza Ali	Roll No. : 67
Mr. Manish Walurkar	Roll No. : 71
Mr. Rohit Farkade	Roll No. : 14
Mr. Mohammad Gaznavi Sheikh	Roll No. : 59
Ms. Mansi Bambode	Roll No. : 03

has completed a **Mini Project** course titled **"COMPUTER NAVIGATION WITH HAND GESTURE RECOGNITION "** towards the partial fulfilment of requirements for the B.E. six semester of Computer Science and Engineering.

Prof. S. W. Shende	Prof. C. U. Chauhan
Project Guide	Project Guide

Prof. S. W. Shende	Prof. C. U. Chauhan
Project Co-ordinator	HOD CSE

DECLARATION

We certify that

a. The work contained in this project has been done by me under the guidance of my supervisor(s).

b. The work has not been submitted to any other Institute for any degree or diploma.

c. We have followed the guidelines provided by the Institute in preparing the project report.

d. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

	Name	Signature
1.	Uzma Ali	
2.	Manish Walurkar	
3.	Rohit Farkade	
4.	Mohammad Gaznavi Sheikh	
5.	Mansi Bambode	

ACKNOWLEDGEMENT

We take this opportunity to express a deep sense of gratitude towards our guide **Prof. Shailendra W. Shende and Prof. Chandrapal U. Chauhan** for providing excellent guidance, encouragement, and inspiration throughout the project work. Without his/her valuable guidance, this work would never have been a successful one. He/she is one of the best mentors, we will always be thankful to him/her.

We would like to extend our special thanks to our Project Coordinator **Prof. Shailendra W. Shende** who has helped us directly or indirectly to complete this project work.

We would like to thank **Prof. Chandrapal U. Chauhan**, Head of the department of Computer Science and Engineering. He was very helpful and encouraging while doing research work.

We would like to thank **Dr. P. V. Washimkar**, Principal (GCOEC) who has provided all institutional facilities as and when needed.

We would also like to thank all our classmates for their valuable suggestions and helpful discussions and our thanks goes to all the people who have supported us to complete the research work directly or indirectly.

Finally, we are extremely grateful to our parents for their love, prayers, care and sacrifices for educating us and preparing us for our future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF Figure

ABSTRACT

The project introduces an application using computer vision for Hand gesture recognition. A camera records a live video stream, from which a snapshot is taken with the help of interface. The system is trained for each type of count hand gestures (one, two, three, four, and five) at least once. After that a test gesture is given to it and the system tries to recognize it.

Hand gestures are a form of nonverbal communication that can be used in several fields such as communication between deaf-mute people, robot control, human–computer interaction (HCI), home automation and medical applications.

Previous systems have used data gloves or markers for input in the system. We have no such constraints for using the system. The user can give hand gestures in view of the camera naturally. A completely robust hand gesture recognition system is still under heavy research and development; the implemented system serves as an extendible foundation for future work.

Keywords: hand gesture; hand posture; computer vision; human–computer interaction (HCI).

CHAPTER 1

INTRODUCTION

1.1 Background:

Computer navigation using hand gesture recognition is an innovative technology that allows users to interact with computers and devices through natural hand movements. This approach enhances user experience by providing a more intuitive and hands-free interaction method compared to traditional input devices.

The scope of this project is to build a real time gesture classification system that can automatically detect gestures in natural lighting condition. In order to accomplish this objective, a real time gesture-based system is developed to identify gestures.

1.2 Key Components:

- **Sensors**: Various sensors, such as cameras or depth sensors, capture and record hand movements in real-time.
- **Gesture Recognition Algorithms**: Advanced algorithms process the captured data to identify and interpret specific hand gestures.
- **User Interface:** The recognized gestures are then translated into commands for the computer, providing a gesture-based interface.

1.3 Methodology

We propose a vision-based approach to accomplish the task of hand gesture detection. As discussed above, the task of hand gesture recognition with any machine learning technique suffers from the variability problem. To reduce the variability in hand recognition task we assume the following assumptions:

- Single coloured camera mounted above a neutral coloured desk.
- User will interact by gesturing in the view of the camera.
- Training is must.
- Hand will not be rotated while image is capturing.
- The real time gesture classification system depends on the hardware and software.

STEPS-

1. It will detect the camera, video interface will be start

2. The camera can extract and recognize human hand gestures from video interface

3. Hand tracking functionality is done by MediaPipe

4. After the recognition the cursor move accordingly, to perform various operations

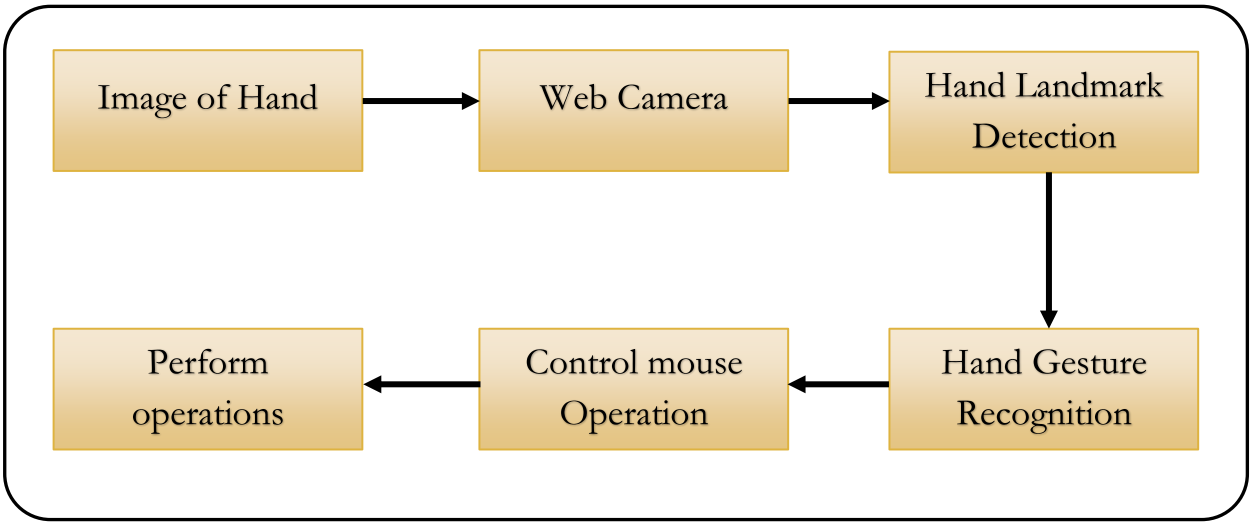


Fig. 1.1: Block Diagram

SOFTWARE TOOLS

Due to the time constraint and complexity of implementing system in PYTHON. The system uses MediaPipe and OpenCV for implementing ML and deep learning techniques to recognize the hand gestures of the user and perform actions. A system that accepted varying inputs of different sizes and image resolutions was implemented; constructing a well coded and documented system for easier future development.

1.4 Objectives

The primary aim of this project is to develop a robust and efficient computer navigation system that utilizes hand gesture recognition technology. The system should enable users to interact with the computer interface and control various functionalities using hand gestures, providing a more intuitive and natural user experience. The specific objectives include:

- **Gesture Recognition System:**
  Implement a reliable and real-time hand gesture recognition system capable of accurately interpreting and classifying a predefined set of gestures.
- **Real-time Performance:**

Optimize the system for real-time performance, minimizing latency between gesture input and system response to ensure a smooth and responsive user experience.

- **Adaptability to applications**:

  Design the system to be adaptable to various user preferences and can be accommodate to various applications based on user needs and preferences.

CHAPTER 02

LITERATURE REVIEW

2.1 Feasibility Report:

A detailed investigation and analysis conducted to determine the financial, economic, technical, or other advisability of a proposed project. Part of the systems development life cycle which aims to determine whether it is sensible to develop some system. The most popular model of feasibility study is "TELOS", standing for Technical, Economic, Legal, Operational, Schedule. Technical Feasibility: The software required to develop the system as well as to run it are easily accessible and free of cost and hence it is technically feasible. Economic Feasibility: The system is simple, user-friendly set up and personalization, cost-efficient/free, easily accessible, and hence it is economically feasible.

Legal Feasibility: There is no conflict between the proposed system and legal requirements.

Feasibility: The current work practices and procedures adequate to support the new system. The system is simple to use, easily accessible and free of cost and hence it is operationally feasible.

Schedule Feasibility: The system is developed within time limit

2.2 Innovations and Usefulness:

1. **Virtual Reality (VR) and Augmented Reality (AR) Navigation:** Hand gesture systems enable users to navigate virtual and augmented reality environments seamlessly. This is especially useful in gaming, training simulations, architectural visualization, and medical training.
2. **Gesture-Controlled Presentations:** Presenters can control slideshows and presentations using hand gestures, allowing for a more dynamic and engaging experience without the need for physical remotes or touchpads.
3. **Medical Imaging and Surgery:** Surgeons can navigate through medical imaging data and control surgical equipment using hand gestures, allowing for precise and intuitive control during minimally invasive procedures.

4. **Gaming and Entertainment:** Gesture-based gaming consoles and entertainment systems offer immersive experiences where players can control characters and interact with virtual environments using natural hand movements, enhancing gameplay and entertainment value.
5. **Industrial Control Systems:** In industrial settings, hand gesture systems can be used to control machinery, robots, and other equipment, improving efficiency and safety by minimizing the need for physical interfaces and reducing the risk of accidents.

2.3 Market Potential and Competitive advantages:

- Market Potential:
  - ✓ Wide Adoption
  - ✓ Technological Advancement
  - ✓ Emerging Markets
  - ✓ Health and Safety Concerns
- Competitive Advantages :
  - ✓ Enhanced Accessibility
  - ✓ Improved Safety
  - ✓ Scalability
  - ✓ Customization

CHAPTER 03

Work Done

3.1 Architecture

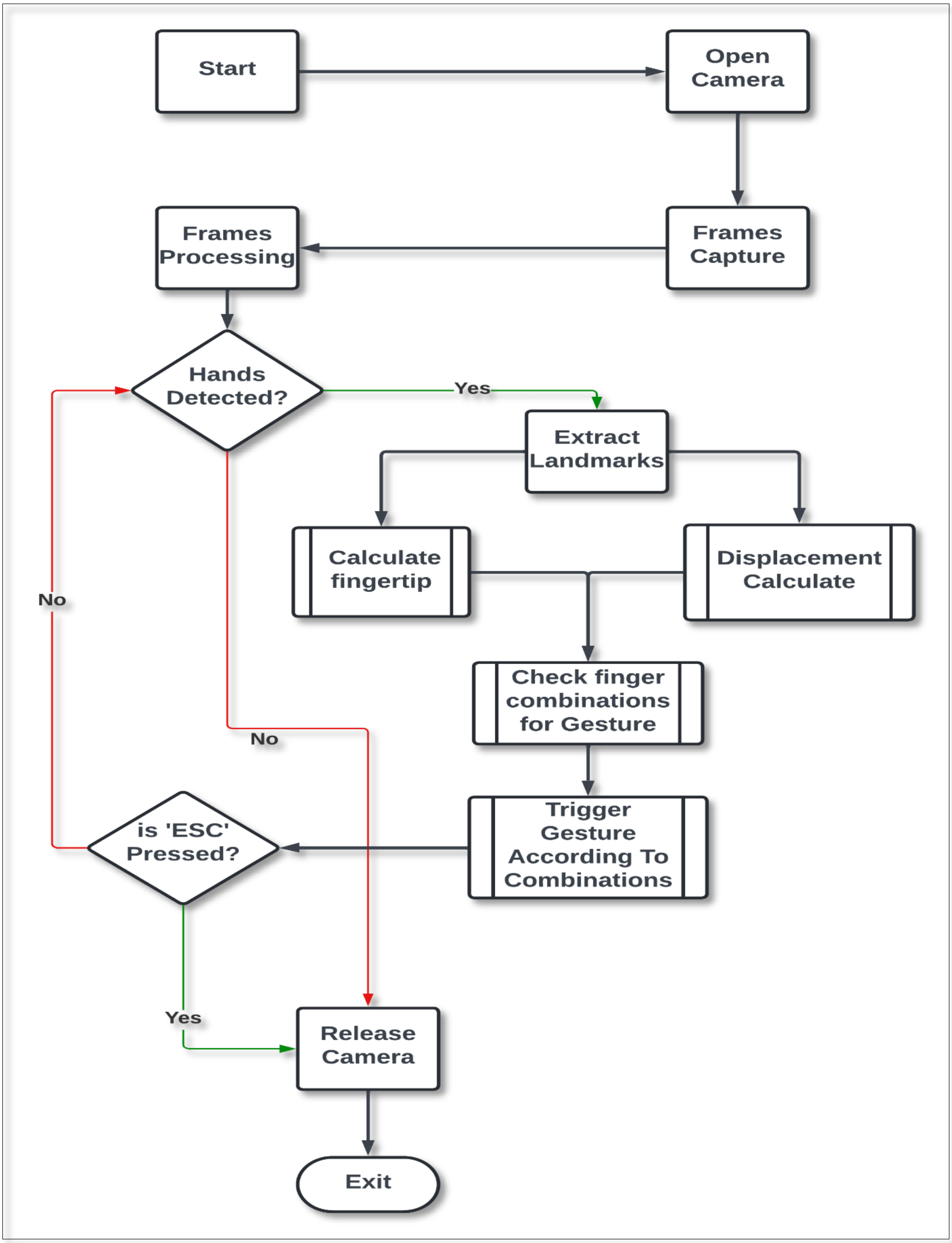


Fig. 3.1: Flow Chart

3.2 Project Constraints

I propose a vision-based approach to accomplish the task of hand gesture detection. As discussed above, the task of hand gesture recognition with any machine learning technique suffers from the variability problem. To reduce the variability in hand recognition task we assume the following assumptions:

- Single coloured camera mounted above a neutral coloured desk.
- User will interact by gesturing in the view of the camera.
- Training is must.
- Hand will not be rotated while image is capturing.
- The real time gesture classification system depends on the hardware and software.

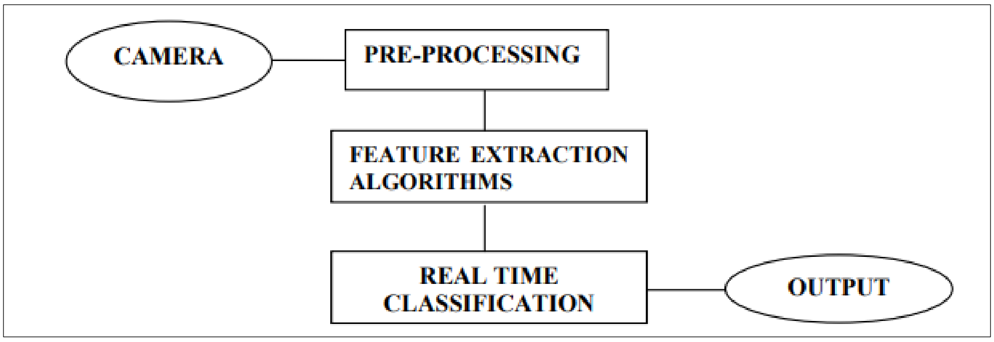


Fig. 3.2: Basic Outline of the Implemented System

3.3 Tools Used

- **Hardware Requirements**
- **Webcam:**
  - ✓ **Specification:** A webcam capable of capturing at least 720p resolution video is recommended for optimal hand gesture detection and tracking.
  - ✓ **Recommended Models:** Logitech C920, Microsoft LifeCam HD-3000, or any webcam with similar or higher specifications.
- **Computer:**
  - ✓ **Processor:** A modern multi-core processor (e.g., Intel Core i5 or AMD Ryzen 5 series) is recommended for real-time video processing and gesture recognition.
  - ✓ **RAM:** At least 4 GB of RAM is recommended for smooth performance.
  - ✓ **Graphics Card:** While not mandatory, a dedicated graphics card can improve overall performance, especially for graphical processing tasks, such as frames processing and more.
  - ✓ **Operating System**: The application can run on Windows, macOS, or Linux platforms.

- **Software Requirements**
  - ✓ **Python (Programming Language):**

    Version: Python 3.6 to Python 3.10. You can download Python from the official Python website and follow the installation instructions based on your operating system.
  - ✓ **OpenCV (Open Source Computer Vision Library):**

    Version: Install the OpenCV library (cv2) using pip: pip install opencv-python. Make sure to use a version compatible with your Python installation (e.g., OpenCV 4.5.x).
  - ✓ **MediaPipe (Google's Machine Learning Framework for ML Solutions):**

    Version: Install the MediaPipe library using pip: pip install mediapipe. Ensure that you have a compatible version of MediaPipe with your Python and OpenCV installations.
  - ✓ **PyAutoGUI (Cross-platform GUI Automation Library for Python):**

    Version: Install PyAutoGUI library using pip: pip install pyautogui. Make sure to use a version compatible with your Python environment.
  - ✓ **Testing Environment:**

    IDEs: Any IDE where the code will be able to execute. Example (JetBrains PyCharm, Microsoft Visual Studio Code, etc)

- **Software for GPU enabling:**
  - ✓ **Install Cuda Toolkit:**

    Download and install the NVIDIA CUDA Toolkit following the instructions provided on the NVIDIA website. During installation, ensure that the CUDA Toolkit is properly configured and added to the system PATH.
  - ✓ **Install cuDNN:**
  - ▪ Download the cuDNN library corresponding to your CUDA Toolkit version from the NVIDIA Developer website.
  - ▪ Follow the cuDNN installation instructions, which usually involve extracting the contents of the downloaded package and copying the necessary files to the CUDA Toolkit directories.
  - ✓ **GPU Driver:**

    Ensure that you have the latest NVIDIA GPU driver installed on your system. The driver should be compatible with both the CUDA Toolkit and cuDNN versions you're using.

- **Operations of cursor-**
  - ✓ Hand Landmark detection
  - ✓ Move
  - ✓ Left Click
  - ✓ Right click
  - ✓ Scroll Up
  - ✓ Scroll down
  - ✓ Volume Control

CHAPTER 04

PROJECT ESTIMATION AND SCHEDULE

4.1 System Estimation Plan

Sr. No	Name	Description	Timeline	Remarks
1	Requirement Analysis.	Complete specification of the system	15-Feb to 28-Feb	A detailed document should be there for each requirement.
2.	High-Level Modelling	Identifies the modules and different entities and their relationships	4-Mar to 2- Mar	Should decide on different modules and how they interact.
3.	Detailed Designing	GUI design, Program Specification	27-Mar to 30-Mar	A detailed designing get done by the team.
4.	Construction	Code for system, Implementation of different modules	13-Feb to 30- Mar	Write code for different modules.
5.	Testing	Test the different modules together	15-Mar to 30-Mar	The system is tested using different test strategies
6.	Deployment	The proposed system get shown to the guide.	5-Apr	All requirements are fulfilled

Table 1: Project Estimation

4.2 Estimate

Project estimation and project scheduling are carried out together without accurate scheduling estimation there is no foundation for effective planning and support for rapid development. There are three parameters involved in computing the total cost of a software development project.

1. Software Cost.

2. Efforts

Our development schedule consists of the following steps.

1. Estimation of the size of the product.

2. Estimation of the effort (Man- months).

3. Estimation of the schedule (Calendar month).

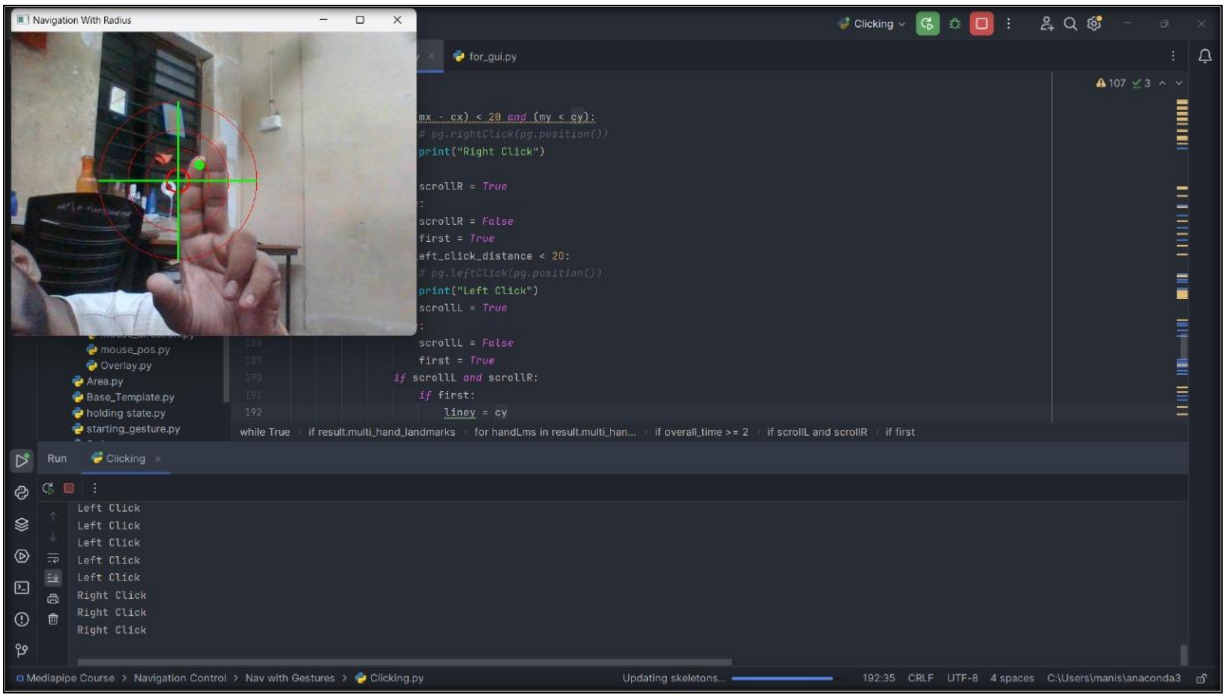
4.3 Schedule

	January		February			March			April		
	15-Jan	28-Jan	3-Feb	13-Feb	27-Feb	4-Mar	15-Mar	30-Mar	5-Apr	12-Apr	20-Apr
Communication	🟨	🟨									
Discussion	🟨										
Requirement Gathering	🟨										
		◆									
Planning			🟪	🟪	🟪	🟪					
Estimate			🟪	🟪							
Scheduling			🟪								
						◆					
Modelling					🟦	🟦	🟦	🟦			
Designing					🟦	🟦	🟦	🟦			
Analysing							🟦	🟦			
Construction				🟦	🟦	🟦	🟦	🟦			
								◆			
Testing							🟥	🟥			
System Testing											
								◆			
Deployment									🟩		
Improvements										🟩	🟩
											◆

Table 2:Schedule (GANT Chart)

CHAPTER 05

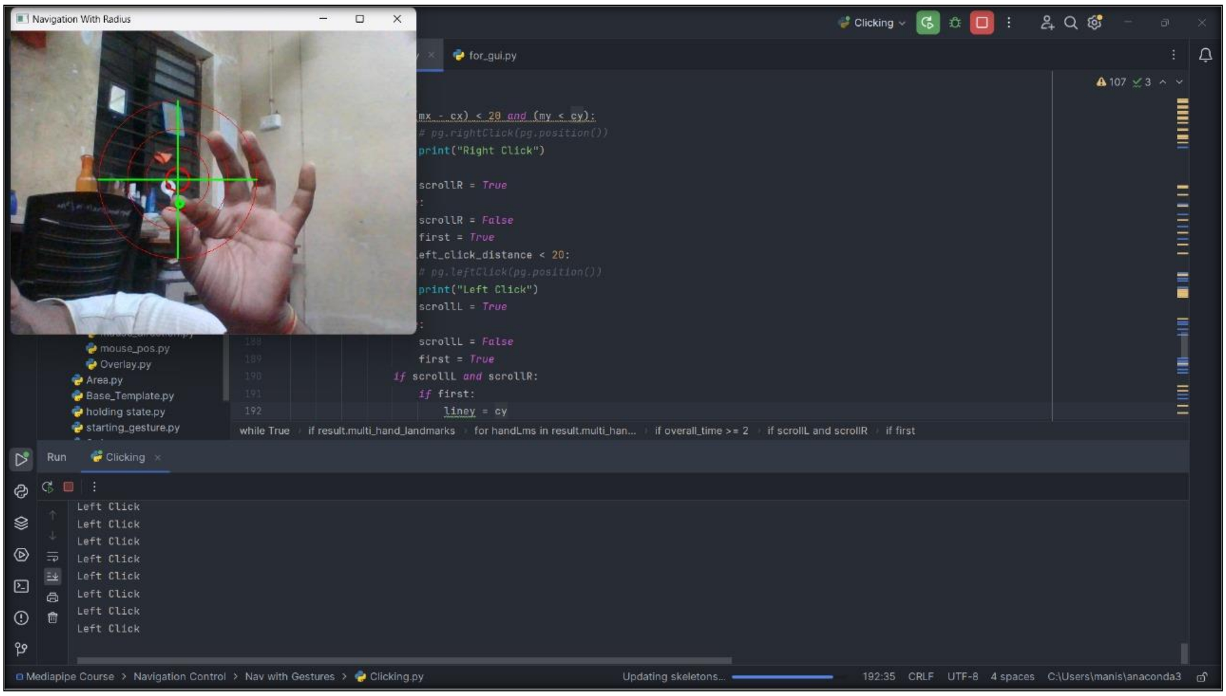
RESULTS AND OUTCOMES


Navigation With Radius
Clicking
for_gui.py
107 3
mouse_pos.py
Overlay.py
Area.py
Base_Template.py
holding state.py
starting_gesture.py
while True › if result.multi_hand_landmarks › for handLms in result.multi_han... › if overall_time >= 2 › if scrollL and scrollR › if first
Run
Clicking
Mediapipe Course › Navigation Control › Nav with Gestures › Clicking.py
Updating skeletons...
192:35 CRLF UTF-8 4 spaces C:\Users\manis\anaconda3

```
mx - cx) < 20 and (my < cy):
    # pg.rightClick(pg.position())
    print("Right Click")

    scrollR = True
:
    scrollR = False
    first = True
eft_click_distance < 20:
    # pg.leftClick(pg.position())
    print("Left Click")
    scrollL = True
:
    scrollL = False
    first = True
if scrollL and scrollR:
    if first:
        liney = cy
```

```
Left Click
Left Click
Left Click
Left Click
Left Click
Right Click
Right Click
Right Click
```

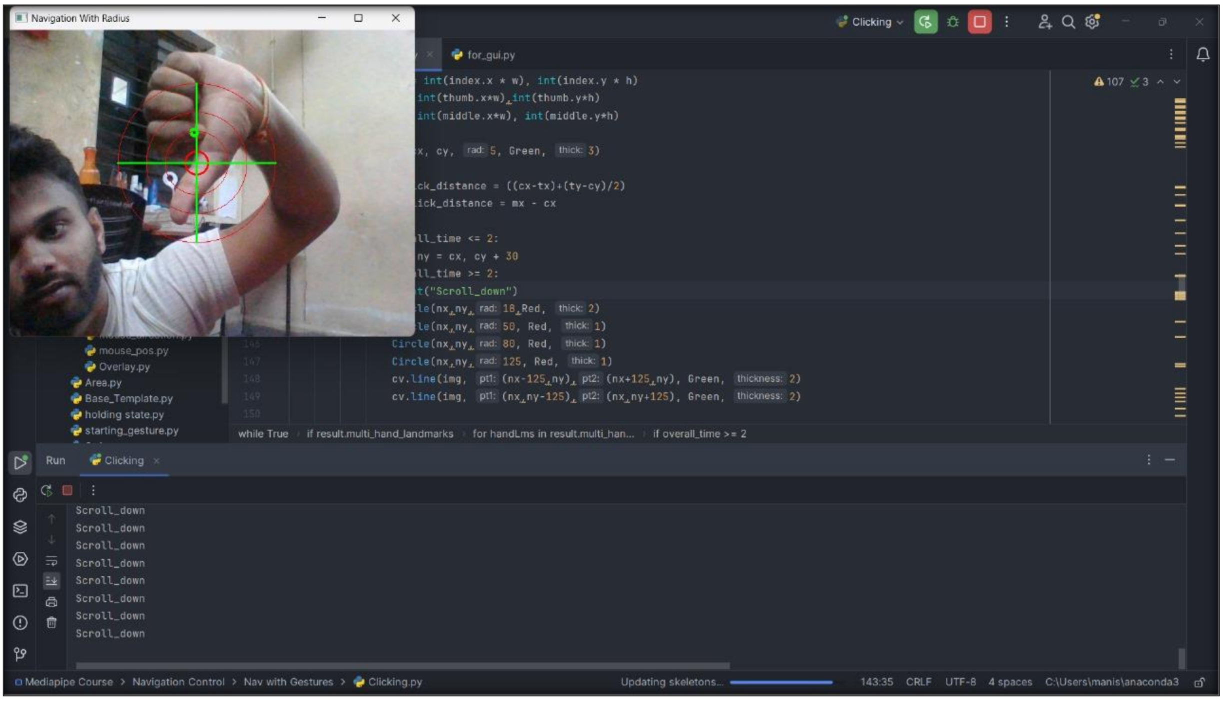
Fig 5.1:Result image of Right Click


Navigation With Radius
Clicking
for_gui.py
107 3
mouse_pos.py
Overlay.py
Area.py
Base_Template.py
holding state.py
starting_gesture.py
while True › if result.multi_hand_landmarks › for handLms in result.multi_han... › if overall_time >= 2 › if scrollL and scrollR › if first
Run
Clicking
Mediapipe Course › Navigation Control › Nav with Gestures › Clicking.py
Updating skeletons...
192:35 CRLF UTF-8 4 spaces C:\Users\manis\anaconda3

```
mx - cx) < 20 and (my < cy):
    # pg.rightClick(pg.position())
    print("Right Click")

    scrollR = True
:
    scrollR = False
    first = True
eft_click_distance < 20:
    # pg.leftClick(pg.position())
    print("Left Click")
    scrollL = True
:
    scrollL = False
    first = True
if scrollL and scrollR:
    if first:
        liney = cy
```

```
Left Click
Left Click
Left Click
Left Click
Left Click
Left Click
Left Click
Left Click
```

Fig 5.2 :Result image of Left Click

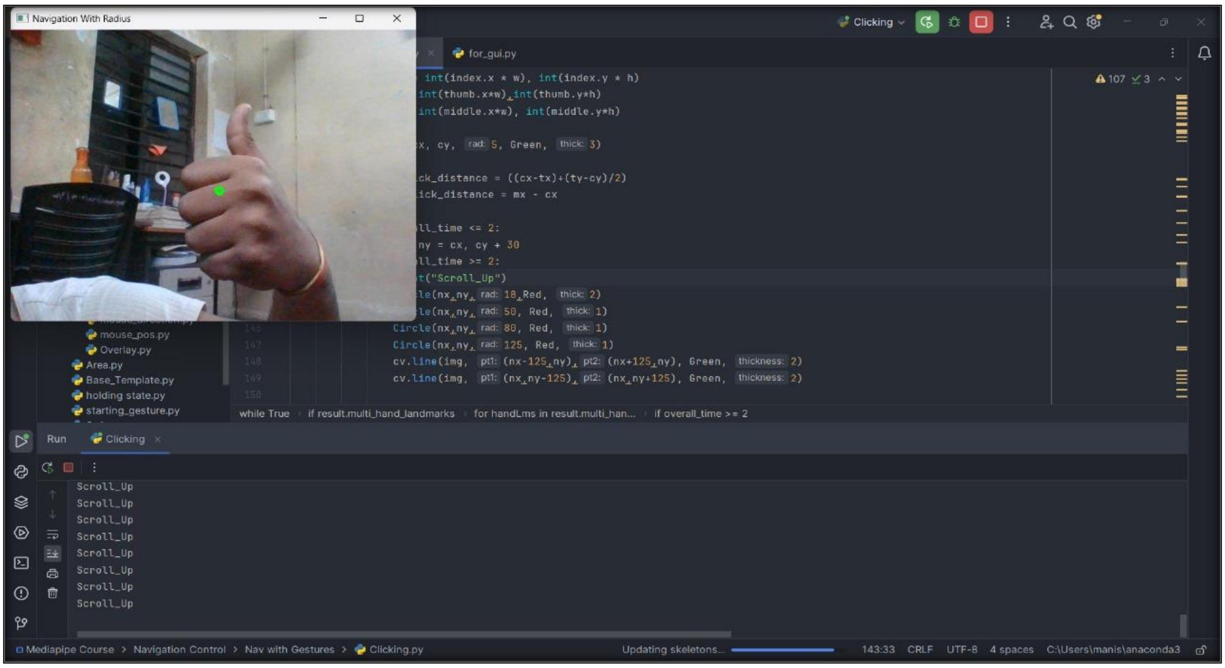


Navigation With Radius
Clicking
for_gui.py
107 3
mouse_pos.py
Overlay.py
Area.py
Base_Template.py
holding state.py
starting_gesture.py
while True › if result.multi_hand_landmarks › for handLms in result.multi_han... › if overall_time >= 2
Run
Clicking
Mediapipe Course › Navigation Control › Nav with Gestures › Clicking.py
Updating skeletons...
143:35 CRLF UTF-8 4 spaces C:\Users\manis\anaconda3

```
 int(index.x * w), int(index.y * h)
int(thumb.x*w),int(thumb.y*h)
int(middle.x*w), int(middle.y*h)

x, cy, rad: 5, Green, thick: 3)

ck_distance = ((cx-tx)+(ty-cy)/2)
ick_distance = mx - cx

ll_time <= 2:
ny = cx, cy + 30
ll_time >= 2:
t("Scroll_down")
le(nx,ny, rad: 18,Red, thick: 2)
le(nx,ny, rad: 50, Red, thick: 1)
Circle(nx,ny, rad: 80, Red, thick: 1)
Circle(nx,ny, rad: 125, Red, thick: 1)
cv.line(img, pt1: (nx-125,ny), pt2: (nx+125,ny), Green, thickness: 2)
cv.line(img, pt1: (nx,ny-125), pt2: (nx,ny+125), Green, thickness: 2)
```

```
Scroll_down
Scroll_down
Scroll_down
Scroll_down
Scroll_down
Scroll_down
Scroll_down
Scroll_down
```

Fig 5.3 :Result image of Scroll Down



Navigation With Radius
Clicking
for_gui.py
107 3
mouse_pos.py
Overlay.py
Area.py
Base_Template.py
holding state.py
starting_gesture.py
while True › if result.multi_hand_landmarks › for handLms in result.multi_han... › if overall_time >= 2
Run
Clicking
Mediapipe Course › Navigation Control › Nav with Gestures › Clicking.py
Updating skeletons...
143:33 CRLF UTF-8 4 spaces C:\Users\manis\anaconda3

```
 int(index.x * w), int(index.y * h)
int(thumb.x*w),int(thumb.y*h)
int(middle.x*w), int(middle.y*h)

x, cy, rad: 5, Green, thick: 3)

ck_distance = ((cx-tx)+(ty-cy)/2)
ick_distance = mx - cx

ll_time <= 2:
ny = cx, cy + 30
ll_time >= 2:
t("Scroll_Up")
le(nx,ny, rad: 18,Red, thick: 2)
le(nx,ny, rad: 50, Red, thick: 1)
Circle(nx,ny, rad: 80, Red, thick: 1)
Circle(nx,ny, rad: 125, Red, thick: 1)
cv.line(img, pt1: (nx-125,ny), pt2: (nx+125,ny), Green, thickness: 2)
cv.line(img, pt1: (nx,ny-125), pt2: (nx,ny+125), Green, thickness: 2)
```

```
Scroll_Up
Scroll_Up
Scroll_Up
Scroll_Up
Scroll_Up
Scroll_Up
Scroll_Up
Scroll_Up
```

Fig 5.4 :Result image of Scroll-Up

CHAPTER 06

FUTURE SCOPE

✓ **Addressing Challenges:** Developing algorithms that can overcome limitations like occlusion (when parts of the hand are hidden) and handle different hand sizes and shapes.

✓ **Combining Vision with Other Sensors:** Fusing data from depth cameras, infrared cameras, and even electromyography (EMG) sensors worn on the arm to provide richer information about hand posture and movement.

✓ **Natural and Intuitive Interfaces:** Creating gesture-based interfaces that are more natural and intuitive for users to interact with computers, virtual reality (VR), and augmented reality (AR) environments.

✓ **Universal Gesture Language:** Developing a standardized set of gestures that can be understood across different cultures and applications, promoting wider adoption.

✓ **Virtual and Augmented Reality:** Enabling seamless interaction with virtual objects and environments using hand gestures for manipulation and control.

✓ **Assistive Technologies:** Providing new communication and control methods for people with disabilities or limitations.

✓ **Sign Language Recognition:** Automatic sign language recognition for improved communication accessibility.

✓ **Automotive and Robotics:** Gesture-based control of vehicles and robots for a more intuitive user experience.

CHAPTER 07

CONCLUSION

Hand gesture recognition addresses a fault in interaction systems. Controlling things by hand is more natural, easier, more flexible and cheaper, and there is no need to fix problems caused by hardware devices, since none is required. From previous sections, it was clear to need to put much effort into developing reliable and robust algorithms with the help of u sing a camera sensor has a certain characteristic to encounter common issues and achieve a reliable result. Each technique mentioned above, however, has its advantages and disadvantages and may perform well in some challenges while being inferior in others.

Hence, we have designed such a system which will replicate the functions performed by hardware devices, effectively and accurately.

CHAPTER 08

REFERENCES

1. https://arxiv.org/ftp/arxiv/papers/1309/1309.2084.pdf
2. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3702844
3. https://www.irjmets.com/uploadedfiles/paper/issue_1_january_2023/33192/final/fin_irjmets1675681559.pdf
4. https://arxiv.org/abs/1512.02325
5. https://arxiv.org/abs/1906.08172

HOD Sign **Project Coordinator**