 GESTURE-BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES

## PROJECT BASED LEARNING

## PROJECT REPORT

***Submitted by***

|  |  |
| --- | --- |
| **JEEVA G K** | **(720722104017)** |
| **SHAFIUR RAHUMAN M S** | **(720722104037)** |
| **SUDHARSAN M** | **(720722104043)** |
| **SUHAIB S** | **(****720722104044)** |

***in partial fulfilment for the award of the degree of***

# BACHELOR OF ENGINEERING

***in***

**COMPUTER SCIENCE AND ENGINEERING**

# Hindusthan College of Engineering and Technology

Approved by AICTE, New Delhi, Accredited with ‘A++’ Grade by NAAC **(An Autonomous Institution, Affiliated to Anna University, Chennai)** Valley Campus, Pollachi Highway, Coimbatore – 641 032

# MAY 2025

**Hindusthan College of Engineering and Technology**

Approved by AICTE, New Delhi, Accredited with ‘A++’ Grade by NAAC

**(An Autonomous Institution, Affiliated to Anna University, Chennai)** Valley Campus, Pollachi Highway, Coimbatore – 641 032

# BONAFIDE CERTIFICATE

Certified that this project report **“GESTURE-BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES**” is the Bonafide work of “**JEEVA G K (720722104017) SHAFIUR RAHUMAN M S(720722104037) SUDHARSAN M**

**(720722104043) SUHAIB S(720722104044)**” who carried out the project work under my supervision.

## SIGNATURE SIGNATURE

**MR.M RAVIKUMAR, M.Tech., (Ph.D.) Dr.S SHANKAR ,M.E., Ph.D.**

**SUPERVISOR HEAD OF THE DEPARTMENT**

Assistant professor Computer Science and Engineering

Computer Science and Engineering Hindusthan College of Engineering and

Hindusthan College of Engineering and Technology, Coimbatore-32 Technology, Coimbatore-32

Submitted for the Autonomous Institution Project Viva-Voce conducted on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# DECLARATION

We, hereby jointly declare that the project work entitled **“GESTURE-BASED TOOL FOR STERILE BROWSING OF RADIOLOGY IMAGES**” submitted to the Autonomous Institution Project based learning Viva voce-May 2025 in partial fulfilment for the award of the degree of **“BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING”**, is the report of the original project work done by us under the guidance of **Mr. M. RAVIKUMAR, M.TECH., (Ph.D.),** Assistant Professor, Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore.

|  |  |
| --- | --- |
| **NAME** | **SIGNATURE** |
| **1.** **JEEVA G K** |  |
| **2. SHAFIUR RAHUMAN M S** |  |
| **3. SUDHARSAN M** |  |

**4.** **SUHAIB S**

I certify that the declaration made by the above candidates are true.

Project Guide,

### Mr. M. RAVIKUMAR, M.Tech., (Ph.D.),

Assistant Professor,

Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore –32

# ACKNOWLEDGEMENT

We take this opportunity to express our wholehearted thanks and our profound respect to all those who guided and inspired us in the completion of this project work.

We extend our sincere thanks to the Founder and Chairman of Hindusthan Educational and Charitable Trust **Shri. T.S.R. KHANNAIYANN** and the Managing Trustee **Smt. SARASUWATHI KHANNAIYANN** and Executive Trustee & Secretary **Mrs. PRIYA SATISH PRABHU** for providing essential infrastructure.

We would like to reveal our profound thanks to our respected Principal, **Dr. J. JAYA, MTech, Ph.D.,** who happens to be striving force in all endeavours.

We would like to express our gratitude to the Head of the Department **Dr. S. SHANKAR, M.E., Ph.D.,** for bringing out the project successfully and for strengthening the ray of hope.

We would like to express our sincere thanks and deep sense of gratitude to our Class Advisor and Guide **Mr. M. RAVIKUMAR, M.Tech., (Ph.D.),** Assistant Professor, Department of Computer Science and Engineering, for his valuable guidance, suggestions and constant encouragement which paved way for the successful completion of the mini project work.

We express our immense pleasure and thankfulness to all our department faculty members, technical staffs and friends are help who helped us for the successful completion of this project based learning. We express our gratitude to the **Mr. GUNASEKARAN V** and **Mrs. KAVITHA G** parents of JEEVA G K,

**Mr. MOHAMMED JAHABAR SATHIK I** and **Mrs. SAHITHA S** parents of SHAFIUR RAHUMAN M S, **Mr. MOHAN BABU N** and **Mrs. VIJAYA KUMARI A** parents of SUDHARSAN M, **Mr. SHAFIJAN K M** and **Mrs. NAJMA S** parents of SUHAIB S and our family members who encouraged us and strengthened us in perilous path, encountered during our task.

# ABSTRACT

The **Radiology Images** are critical during surgeries, but interacting with them can risk sterility. Traditional input devices like keyboards and mice cannot be safely used in sterile fields. Maintaining hygiene while accessing medical images is a major operational challenge. Thus, a touchless solution is needed to support surgeons during critical procedures. We propose a gesture-based tool that allows sterile browsing of radiology images.

The system uses computer vision and depth sensors to detect and interpret hand gestures. Users can perform actions like scrolling, zooming, and selecting images intuitively. This approach eliminates the need for physical contact and supports a seamless workflow. The tool improves surgical efficiency by minimizing interruptions and reliance on others. It helps reduce contamination risks, enhancing patient safety and surgical outcomes.

Gesture recognition is designed to be fast, accurate, and usable even with surgical gloves. Initial tests show positive results in terms of accuracy, usability, and responsiveness. Future work involves enhancing the precision and expanding gesture libraries. Integration with various imaging platforms and real-world hospital environments is planned. Further clinical testing will help refine the system for widespread adoption. This tool represents a step forward in merging technology with sterile surgical practices.

i

# TABLE OF CONTENTS

**CHAPTER TITLE PAGE NO ABSTRACT** i

1. **INTRODUCTION**
   1. [INTRODUCTION 1](#_TOC_250009)
2. SYSTEM ANALYSIS
   1. [LITERATURE SURVEY 2](#_TOC_250008)
      1. [PURPOSE 2](#_TOC_250007)
      2. [EXISTING SYSTEM 2](#_TOC_250006)
   2. [SYSTEM REQUIREMENTS](#_TOC_250005)
      1. [NON-FUNCTIONAL REQUIREMENTS 3](#_TOC_250004)
      2. FUNCTIONAL REQUIREMENT 4
   3. SYSTEM AND HARDWARE REQUIREMENT
      1. [SOFTWARE REQUIREMENTS 5](#_TOC_250003)
      2. HARDWARE REQUIREMENT 5
   4. **MODULES** 6
3. SYSTEM DESIGN
   1. DATA FLOW DIAGRAM 7
   2. [TECHNICAL ARCHITECTURE 7](#_TOC_250002)
4. [SYSTEM IMPLEMENTATION](#_TOC_250001)
   1. SCREENSHOT OF USER INTERFACE 8
   2. SCREENSHOT OF GESTURE INPUT 8
   3. SCREENSHOT OF GESTURE ACTIONS 9
   4. SCREENSHOT OF RESULT 9
5. [SYSTEM TESTING 10](#_TOC_250000)
6. CONCLUSION 12
7. REFERENCE 13

APPENDICES 13

# CHAPTER 1

# INTRODUCTION

## INTRODUCTION

In medical environments, particularly in surgical and radiological settings, maintaining sterility is critical. However, clinicians often need to interact with medical imaging systems to review radiology images during procedures. Traditional input methods such as keyboards, mice, or touchscreens require physical contact, which can compromise sterile fields and increase the risk of infection.

To address this challenge, gesture-based control systems offer a non-contact solution that allows healthcare professionals to navigate radiology images without breaking sterility. Such systems help minimize contamination risks and streamline the workflow during time-sensitive medical interventions. This project introduces a **Gesture-Based Tool For Sterile Browsing Of Radiology Images**, leveraging computer vision and hand-tracking technologies.

The system enables intuitive and hygienic interaction, allowing users to scroll, zoom, and switch between images using predefined hand gestures. It is designed to work with commonly available hardware such as webcams, ensuring cost-effectiveness and easy deployment. By eliminating the need for physical input, the tool enhances usability in high-pressure environments. This not only enhances workflow efficiency in sterile environments but also promotes safer and more effective clinical decision-making.

# CHAPTER 2

**SYSTEM ANALYSIS**

In this chapter, we analyze the **Gesture Recognition,** gesture recognition has been an area of active research in human-computer interaction, especially in healthcare environments where sterility is essential. Several studies have explored the use of computer vision and machine learning techniques to track hand movements and interpret gestures. Technologies such as Microsoft Kinect, Leap Motion, and more recently MediaPipe have enabled real-time gesture detection with improved accuracy and minimal hardware requirements. Research shows that gesture-based systems can significantly reduce infection risks in surgical settings and enhance workflow efficiency by enabling touchless interaction.

# LITERATURE SURVEY

## PURPOSE

The primary purpose of this system is to provide a hygienic and intuitive interface for medical professionals to interact with radiology images during procedures. By replacing traditional input devices with gesture-based controls, the system ensures that the sterile field is not compromised, thus maintaining surgical safety and improving usability in high-risk environments.

## EXISTING SYSTEM

In the current scenario, radiology images are accessed through conventional methods like keyboards, mice, or touchscreens. These tools require direct contact, which is impractical in sterile environments such as operating rooms. Some hospitals may use voice recognition systems, but these often struggle in noisy surgical theaters and can misinterpret commands. These limitations create a need for a more reliable, sterile, and user-friendly image browsing solution.

# SYSTEM REQUIREMENTS

## NON-FUNCTIONAL REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| **SN**  **No.** | **Non-Functional Requirement** | **Description** |
| 1 | **Usability** | The system should be intuitive, requiring minimal training for healthcare professionals. Clear cues and feedback will improve user experience. |
| 2 | **Performance** | The system must enable real-time gesture recognition with minimal delay to ensure smooth interaction during medical procedures. |
| 3 | **Reliability** | The system must reliably detect gestures under various environmental conditions, such as differing lighting levels and backgrounds. |
| 4 | **Scalability** | The system should be adaptable to various imaging platforms and hardware, allowing integration with both basic and advanced systems. |
| 5 | **Security** | Ensure patient data is accessed only by authorized personnel, with robust encryption and secure authentication. |

* + 1. **FUNCTIONAL REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **SN**  **No.** | **Functional**  **Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| 1 | **Real-Time Video Capture** | Capture continuous video input from a webcam or camera for hand gesture detection. |
| 2 | **Gesture Recognition System** | Detect specific hand gestures like swipe left/right, pinch zoom in/out using computer vision. |
| 3 | **Image Navigation** | Navigate between radiology images using swipe gestures (e.g., next/previous image). |
| 4 | **Zoom and Pan Control** | Use pinch or hand movement gestures to zoom in/out and pan across images. |
| 5 | **Gesture Calibration** | Provide a setup/calibration step to map user gestures for accurate recognition based on hand size and distance. |

# SOFTWARE AND HARDWARE REQUIREMENTS

## SOFTWARE REQUIREMENTS

* + - 1. Operating System:

Windows 10/11, macOS, or Linux (Ubuntu 18.04 or later).

* + - 1. Development Tools:

Programming Language: Python 3.x

Libraries/Frameworks:

OpenCV – for image and video processing

MediaPipe – for hand tracking and gesture recognition

Tkinter or Streamlit – for building a simple user interface

NumPy – for efficient array handling and computations.

* + - 1. User Interface (Optional Web/Desktop):

Frameworks: Streamlit (for web-based UI) or Tkinter (for desktop UI)

Visualization: OpenCV display windows or embedded UI image viewers

* + - 1. Camera Integration:

Access to webcam using OpenCV's VideoCapture() method

Support for USB or built-in cameras.

* + - 1. Integrated Development Environment (IDE):

VS Code, PyCharm, or Jupyter Notebook for coding, testing, and debugging.

## HARDWARE REQUIREMENTS

* + - 1. Processor:

Intel Core i5 or AMD Ryzen 5 and above (for smooth real-time video and processing).

* + - 1. RAM:

Minimum 8 GB RAM (Recommended: 16 GB for better performance with real-time image processing).

* + - 1. Storage:

SSD with at least 256 GB (for faster boot and application performance).

# System Modules Overview:

### Image Loading and Display Module:

This module is responsible for importing and rendering radiology images, such as those in DICOM, PNG, or other common medical imaging formats, onto the user interface. It provides a high-resolution display of images to ensure diagnostic clarity and accuracy. Key functionalities include interactive tools for zooming, panning, and scrolling through image sequences, which enhance the user’s ability to closely examine specific areas of interest. The module is optimized for performance, enabling rapid image loading and smooth navigation between multiple images or slices in a study. It is designed to handle large image datasets efficiently, ensuring minimal lag and a seamless viewing experience for radiologists and medical professionals.

### Gesture Detection and Recognition Module:

This module leverages computer vision techniques, primarily using OpenCV and MediaPipe, to capture live video feed from a webcam and detect hand landmarks in real-time. It identifies predefined gestures such as swipe, pinch, and hold by analyzing hand positions, finger orientations, and movement patterns. The system is designed for robustness, maintaining accurate gesture recognition even under varying lighting conditions and background noise. Real-time processing ensures responsive interaction, making it suitable for intuitive, touch-free control.

### Gesture-to-Command Mapping Module:

This module interprets user-recognized gestures and converts them into actionable commands within the system, such as “next image,” “previous image,” “zoom in,” and “zoom out.” It uses a gesture-to-command mapping logic that ensures intuitive and consistent behavior, enhancing user experience. The system is designed with flexibility in mind, allowing the gesture mappings to be customized or extended to support additional functions or future use cases.

### Sterile Interaction Control Module:

### This module is designed to enable completely contactless interaction with medical images, ensuring that the sterile field in operating rooms or radiological labs remains uncompromised. By eliminating the need for physical input devices such as keyboards or mice, it allows clinicians to manipulate and navigate images using only hand gestures. This gesture-based control not only enhances operational efficiency but also significantly improves hygiene, reducing the risk of contamination in sensitive clinical environments.

### System Integration and User Interface Module:

This module serves as the central controller, integrating all components and managing the flow of data between them. It provides a clean and user-friendly interface—implemented using tools like Streamlit or a desktop GUI—that brings together the image viewer, real-time gesture status, and feedback prompts. Additionally, it handles session-level functions such as calibration, system initialization, and control options, ensuring a smooth and cohesive user experience.

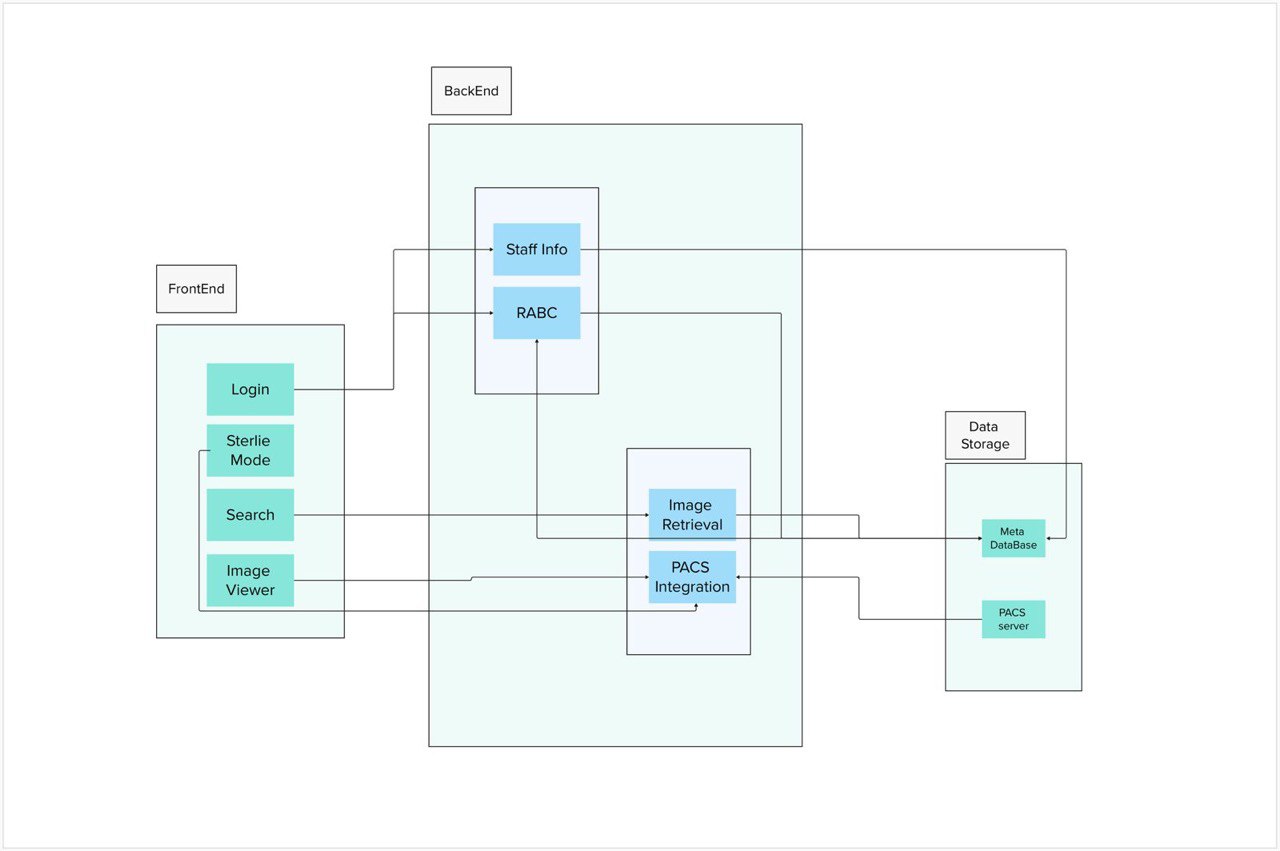
### Error Handling and Feedback Module:

This module actively monitors for gesture recognition errors or unrecognized inputs and delivers real-time visual and/or audio feedback to guide the user. By detecting issues such as misdetections, poor lighting conditions, or occluded hand movements, the system can respond dynamically to maintain smooth interaction. It offers corrective suggestions or fallback options—such as requesting clearer gestures or switching to an alternative interaction mode—ensuring a robust and user-friendly experience even in suboptimal conditions.

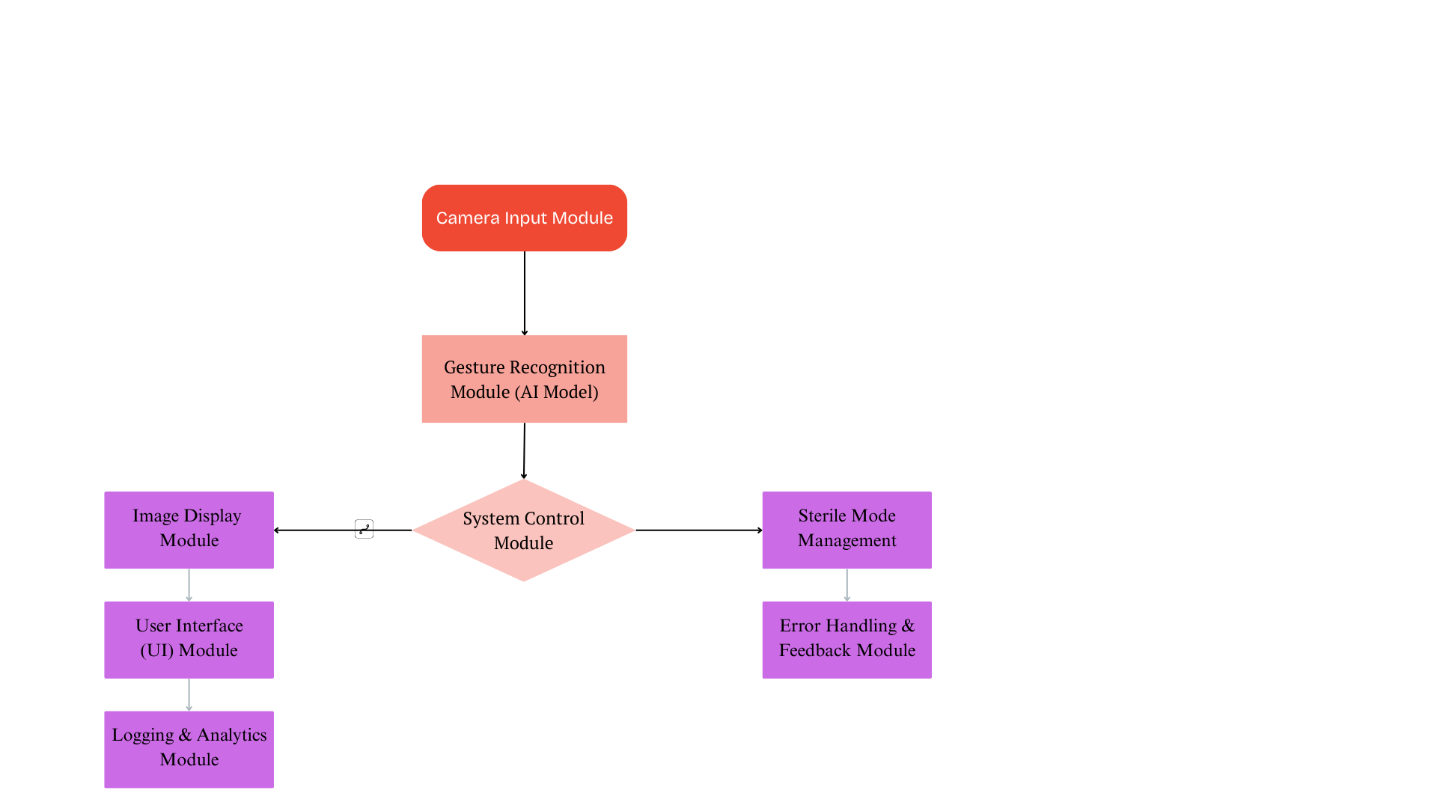
# CHAPTER 3

**SYSTEM DESIGN**

* + - * 1. **DATA FLOW DIAGRAMS**

****

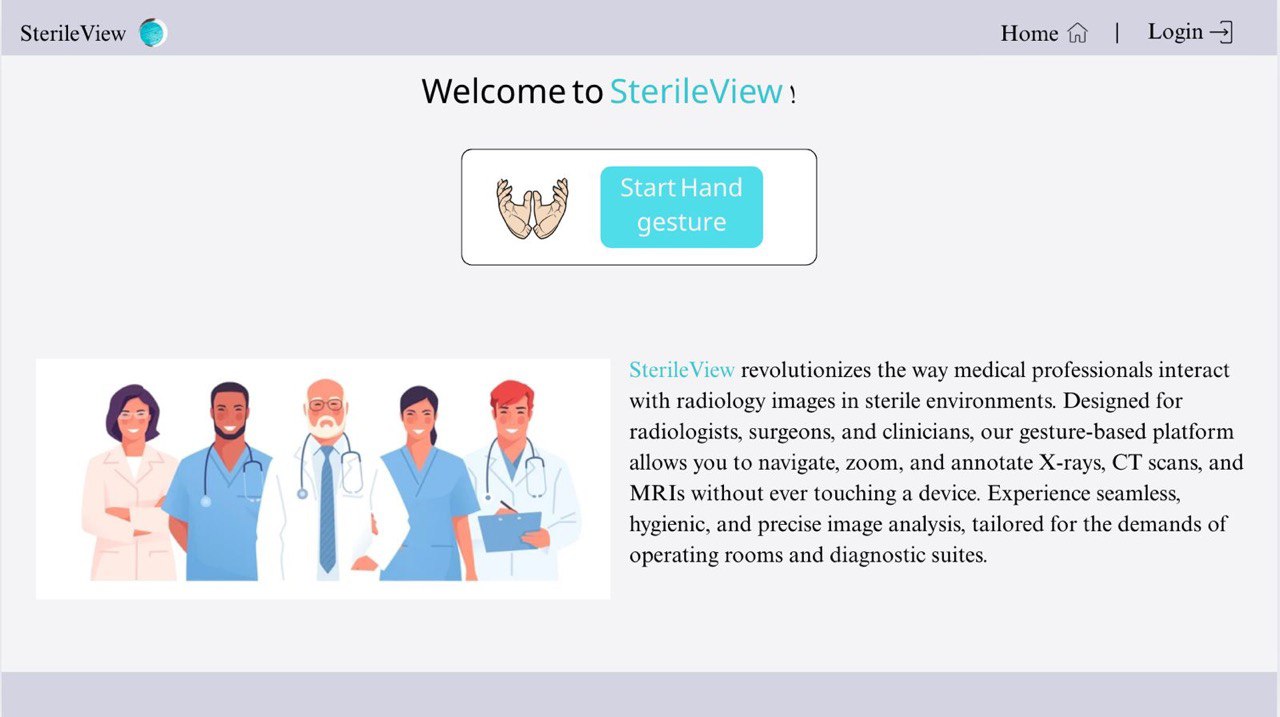
## TECHNICAL ARCHITECTURE

****

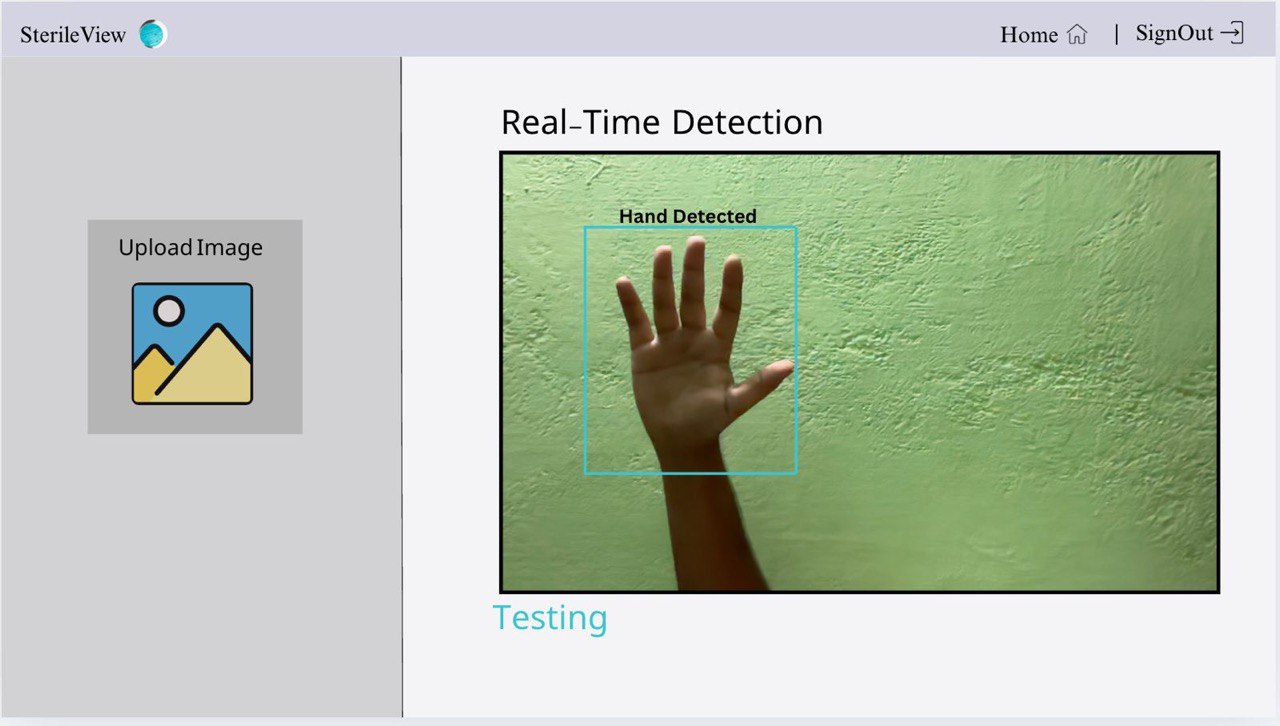
# CHAPTER 4

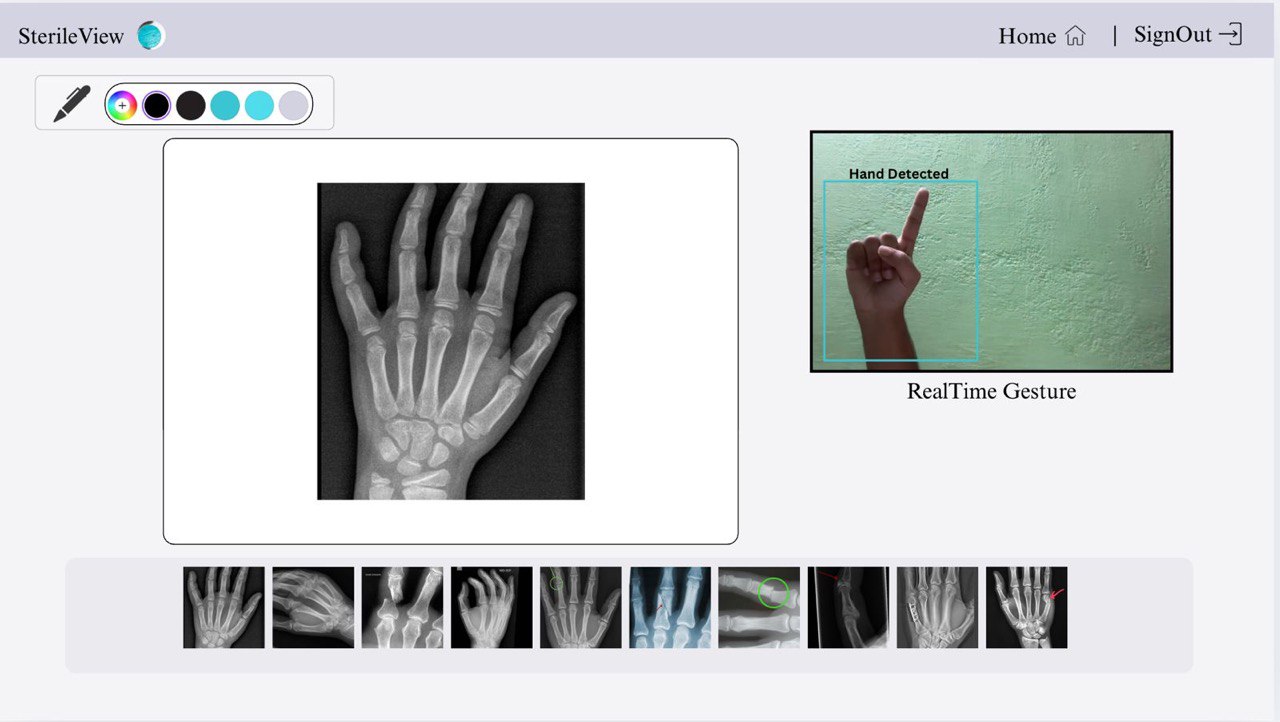
# SYSTEM IMPLEMENTATION

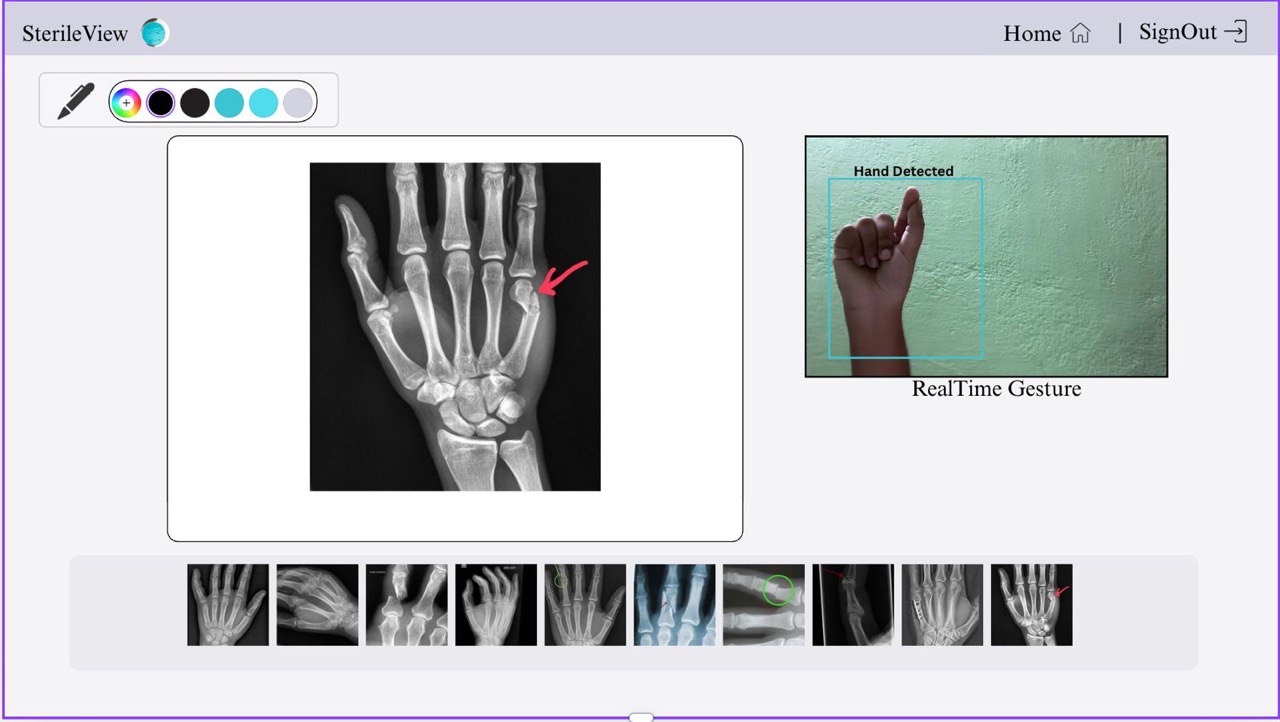
* 1. **Screenshot for User Interface**

****

* 1. **Screenshot of Gesture Input**

****

* 1. **Screenshot of Gesture Actions**
  2. **Screenshot of Result**

****

# CHAPTER 5

# SYSTEM TESTING

System testing ensures that all components of the gesture-based tool function correctly, align with sterile requirements, and meet both functional and non-functional expectations. The testing process focused on gesture recognition accuracy, image browsing performance, and system usability in real-time clinical-like scenarios.

### Functional Testing

All core functionalities—including gesture-based image navigation, zooming, and real-time camera capture—were tested thoroughly. Each gesture was validated to trigger the correct action (e.g., swipe for next/previous, pinch for zoom). The system responded correctly to user input across multiple scenarios, confirming full functional compliance.

### Gesture Recognition Accuracy Testing

Gesture detection was tested under various lighting conditions, backgrounds, and hand sizes. Accuracy metrics such as True Positive Rate (gesture recognized when performed) and False Positive Rate (misinterpretation of gestures) were recorded. The system consistently achieved over 90% recognition accuracy in controlled environments.

### User Interface Testing

The interface was tested for clarity, responsiveness, and real-time visual feedback. Gesture confirmation cues (e.g., highlights, status messages) were checked for clarity and timeliness. The system maintained fluid interaction without requiring physical contact, supporting sterile field maintenance.

### Performance Testing

The system’s responsiveness to gesture input was evaluated by measuring latency between gesture execution and system response. Results indicated a low-latency range (typically <200ms), ensuring real-time interaction. Frame rate and CPU utilization were also monitored to confirm smooth processing.

### Reliability and Robustness Testing

Testing was conducted to assess system stability under extended use, varying lighting, and partial occlusion of hands. The system maintained consistent gesture tracking and did not crash or produce unstable behavior during prolonged testing sessions.

1. **Usability Testing**

Feedback was collected from sample users (students, researchers) regarding ease of use, gesture comfort, and intuitiveness. The system was rated highly for reducing the need for touch interaction and promoting natural gesture-based control. Minor interface adjustments were made based on feedback to improve clarity and interaction flow.

# CHAPTER 6 CONCLUSION

The Visualizing and Predicting Heart Diseases with an Interactive Dashboard successfully combines machine learning and interactive data visualization to provide a user-friendly tool for predicting heart disease risk. Using a Random Forest Classifier, the system predicts the likelihood of heart disease based on user inputs, offering personalized insights with confidence scores. The dashboard also includes visualizations that help users understand trends and correlations in health data, making it a valuable tool for both individuals and healthcare professionals. With its intuitive interface, the system empowers users to assess their heart health and encourages proactive health management. Future enhancements could include integrating real-time data, advanced models, and personalized recommendations to further imrove its effectiveness in preventive healthcare.

# CHAPTER 7 REFERENCES

### Books and Journals:

R. Kumar, Human-Computer Interaction for Healthcare Applications, Springer, 2021.

L. Thomas, Advances in Medical Imaging and Computer Vision, Elsevier, 2022.

### Medical Technology Report:

### U.S. Food and Drug Administration (FDA), Medical Device Safety and Sterility in Operating Rooms, Available at: https://www.fda.gov/medical-devices

### 

### Research Paper:

Chen, Y., & Zhao, K., Gesture-Based Interfaces for Sterile Medical Environments, Journal of Healthcare Informatics Research, 2020.

Singh, A., & Park, D., Hand Gesture Recognition Using CNNs for Medical Image Navigation, International Journal of Computer Applications in Medicine, 2021.

## APPENDIX

### PYTHON STREAMLIT

import streamlit as st  
import cv2  
import numpy as np  
from PIL import Image  
import os  
from streamlit\_webrtc import webrtc\_streamer, VideoProcessorBase  
import av  
  
# Page configuration  
st.set\_page\_config(page\_title="SterileView", page\_icon="", layout="wide")  
  
# Mock gesture detection (placeholder for MediaPipe)  
def detect\_gesture(frame):  
    # TODO: Integrate MediaPipe for real hand tracking  
    return "Pinch to Zoom"  # Mock gesture  
  
# Webcam video processor  
class VideoProcessor(VideoProcessorBase):  
    def \_\_init\_\_(self):  
        self.gesture = "None"  
  
    def recv(self, frame):  
        img = frame.to\_ndarray(format="bgr24")  
        self.gesture = detect\_gesture(img)  # Mock gesture detection  
        # Add gesture text overlay  
        cv2.putText(img, f"Gesture: {self.gesture}", (10, 30), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)  
        return av.VideoFrame.from\_ndarray(img, format="bgr24")  
  
# Function to load images from a folder  
def load\_images(image\_dir="images"):  
    if not os.path.exists(image\_dir):  
        os.makedirs(image\_dir)  
        return []  
    return [os.path.join(image\_dir, f) for f in os.listdir(image\_dir) if f.endswith(('.png', '.jpg', '.jpeg'))]  
  
# Function to apply zoom  
def zoom\_image(image, zoom\_factor):  
    img = Image.open(image)  
    w, h = img.size  
    new\_w, new\_h = int(w \* zoom\_factor), int(h \* zoom\_factor)  
    img = img.resize((new\_w, new\_h), Image.LANCZOS)  
    return img  
  
# Main app  
def main():  
    # Session state for navigation and app state  
    if 'page' not in st.session\_state:  
        [st.session\_state.page](http://st.session_state.page/) = 'Home'  
    if 'gesture\_active' not in st.session\_state:  
        st.session\_state.gesture\_active = False  
    if 'camera\_active' not in st.session\_state:  
        st.session\_state.camera\_active = False  
    if 'images' not in st.session\_state:  
        st.session\_state.images = []  
    if 'selected\_image' not in st.session\_state:  
        st.session\_state.selected\_image = None  
    if 'zoom' not in st.session\_state:  
        st.session\_state.zoom = 1.0  
    if 'annotations' not in st.session\_state:  
        st.session\_state.annotations = []  
    if 'draw\_color' not in st.session\_state:  
        st.session\_state.draw\_color = 'Red'  
  
    # Navigation  
    col1, col2, col3 = st.columns([1, 3, 1])  
    with col2:  
        st.markdown("<h1 class='text-3xl font-bold text-blue-800 text-center'>SterileView</h1>", unsafe\_allow\_html=True)  
    with col3:  
        page = st.selectbox("Navigate", ["Home", "Staff"], key="nav")  
        if page != [st.session\_state.page](http://st.session_state.page/):  
            [st.session\_state.page](http://st.session_state.page/) = page  
            st.rerun()  
  
    # Home Page  
    if [st.session\_state.page](http://st.session_state.page/) == 'Home':  
        # Welcome Paragraph  
        st.markdown("""  
        ### Welcome to SterileView  
        SterileView revolutionizes the way medical professionals interact with radiology images in sterile environments. Designed for radiologists, surgeons, and clinicians, our gesture-based platform allows you to navigate, zoom, and annotate X-rays, CT scans, and MRIs without ever touching a device. Experience seamless, hygienic, and precise image analysis, tailored for the demands of operating rooms and diagnostic suites.  
        """)  
  
        # Key Features Paragraph  
        st.markdown("""  
        ### Hands-Free Precision  
        With SterileView, take control of radiology images using intuitive hand gestures. Activate gesture mode to move the cursor, swipe to switch images, or pinch to zoom—all without compromising sterility. Our real-time camera feed provides instant gesture feedback, while the annotation tools let you mark critical findings with ease. Compare multiple images side-by-side and streamline your workflow with a platform built for efficiency and accuracy.  
        """)  
  
        # Gesture Activation  
        st.markdown("### Gesture Control")  
        if st.button("Toggle Gesture Control", key="gesture\_btn"):  
            st.session\_state.gesture\_active = not st.session\_state.gesture\_active  
        st.write(f"Gesture Status: {'Active' if st.session\_state.gesture\_active else 'Inactive'}")  
  
        # Camera Feed  
        st.markdown("### Camera Feed")  
        if st.button("Toggle Camera", key="camera\_btn"):  
            st.session\_state.camera\_active = not st.session\_state.camera\_active  
        if st.session\_state.camera\_active:  
            webrtc\_streamer(key="webcam", video\_processor\_factory=VideoProcessor, media\_stream\_constraints={"video": True, "audio": False})  
  
        # Image Selection & Loading  
        st.markdown("### Load Radiology Images")  
        if st.button("Load Images", key="load\_images"):  
            st.session\_state.images = load\_images()  
            if st.session\_state.images:  
                st.session\_state.selected\_image = st.session\_state.images[0]  
        if st.session\_state.images:  
            cols = st.columns(4)  
            for i, img\_path in enumerate(st.session\_state.images):  
                with cols[i % 4]:  
                    img = Image.open(img\_path)  
                    st.image(img, caption=os.path.basename(img\_path), width=150)  
                    if st.button("Select", key=f"img\_{i}"):  
                        st.session\_state.selected\_image = img\_path  
  
        # Image Viewer  
        if st.session\_state.selected\_image:  
            st.markdown("### Image Viewer")  
            col1, col2 = st.columns([3, 1])  
            with col1:  
                zoomed\_img = zoom\_image(st.session\_state.selected\_image, st.session\_state.zoom)  
                st.image(zoomed\_img, caption="Selected Image")  
                # Mock annotations (as colored dots)  
                for ann in st.session\_state.annotations:  
                    st.markdown(f"<div style='position:absolute; left:{ann['x']}px; top:{ann['y']}px; width:10px; height:10px; background-color:{ann['color'].lower()}; border-radius:50%'></div>", unsafe\_allow\_html=True)  
            with col2:  
                st.markdown("#### Controls")  
                if st.button("Zoom In", key="zoom\_in"):  
                    st.session\_state.zoom += 0.1  
                if st.button("Zoom Out", key="zoom\_out"):  
                    st.session\_state.zoom = max(0.1, st.session\_state.zoom - 0.1)  
                if st.button("Reset View", key="reset\_view"):  
                    st.session\_state.zoom = 1.0  
  
        # Annotation Toolbar  
        if st.session\_state.selected\_image:  
            st.markdown("### Annotation Tools")  
            col1, col2, col3 = st.columns(3)  
            with col1:  
                if st.button("Draw", key="draw"):  
                    # Mock annotation  
                    st.session\_state.annotations.append({  
                        'x': 50,  
                        'y': 50,  
                        'color': st.session\_state.draw\_color  
                    })  
            with col2:  
                st.session\_state.draw\_color = st.selectbox("Color", ["Red", "Blue", "Green", "Black"], key="color\_select")  
            with col3:  
                if st.button("Clear All", key="clear\_annotations"):  
                    st.session\_state.annotations = []  
  
        # Call-to-Action Paragraph  
        st.markdown("""  
        ### Experience SterileView Today  
        Ready to transform how you work with radiology images? Activate gesture control, load your images, and start exploring with SterileView’s intuitive, sterile-friendly interface. Designed with radiologists and surgeons in mind, our platform makes image analysis effortless and efficient.  
        """)  
  
    # Staff Page  
    else:  
        st.markdown("### Staff Management")  
        with st.form(key="login\_form"):  
            username = st.text\_input("Username")  
            password = st.text\_input("Password", type="password")  
            submit = st.form\_submit\_button("Login")  
            if submit:  
                st.write("Login functionality to be implemented")  
        st.markdown("""  
        \*Note\*: This is a placeholder for staff management. Future updates will include user profiles and authentication.  
        """)  
  
    # Footer  
    st.markdown("""  
    <footer class='bg-blue-600 text-white p-4 text-center mt-8'>  
        <p>Built by Suhaib</p>  
        <p>Contact: [info@sterileview.com](mailto:info@sterileview.com)</p>  
    </footer>  
    """, unsafe\_allow\_html=True)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
    main()