Cursor Control of Desktop Application using Coloured Finger Gesture

---

A report submitted for the J component of
CSE4015 - Human Computer Interaction

Supervisor
Prof. Perepi Rajarajeswari

School of Computer Science and Engineering
Vellore Institute of Technology, Vellore

April, 2023

Team Details:

Name	**Reg No.**	**College Email ID**	**Contact No.**
Bhuwan Rathi	20BCE2808	bhuwan.rathi2020@vitstudent.ac.in	8420892833
Biggyat Kumar Pandey	20BCE2763	biggyatkumar.pandey2020@vitstudent.ac.in	8807696269
Baibhav Rijal	20BCE2779	baibhav.rijal2020@vitstudent.ac.in	7586948551

Declaration

We hereby declare that the project/J-component entitled "**Cursor Control of Desktop Application using Coloured Finger Gesture**" submitted to Vellore Institute of Technology, Vellore, is a record of an original work done by **Bhuwan Rathi (20BCE2808), Biggyat Kumar Pandey (20BCE2763), Baibhav Rijal (20BCE2779)** under the guidance of **Prof. Perepi RajaRajeswari**, SCOPE, Vellore Institute of Technology during the academic year 2023. This report has been prepared on the basis of our own work with the help of some published and unpublished source materials, which have been acknowledged.

Student Names: Bhuwan Rathi(20BCE2808)
Biggyat Kumar Pandey (20BCE2763)
Baibhav Rijal (20BCE2779)

Place: **Vellore Institute of Technology**

Submission: **April 2023**

Acknowledgement

It is with great pleasure that we extend our sincere gratitude and appreciation to each one of us for our contribution to the completion of this group project report.

First and foremost, we would like to thank each member of our group for their hard work, dedication, and commitment to this project. The success of this report would not have been possible without the collaborative effort of all of us.

We would also like to express our gratitude to our supervisor **Prof. Perepi Rajarajeswari**, whose guidance and support were crucial in shaping our ideas and ensuring the successful completion of this project. Their constructive feedback and suggestions were invaluable in improving the quality of our report.

We would also like to acknowledge the support of the staff faculty of **Vellore Institute of Technology**, who provided us with the necessary resources and facilities to undertake this project. Their support and encouragement were a source of inspiration and motivation for us.

Once again, we extend our heartfelt gratitude to everyone who has contributed to this project report. Without your support and collaboration, this would not have been possible.

Sincerely,
Bhuwan Rathi(20BCE2808)
Biggyat Kumar Pandey (20BCE2763)
Baibhav Rijal (20BCE2779)

Abstract

This project aims to develop a novel approach for controlling desktop applications using finger gestures. The system will utilise a webcam to detect coloured gloves worn by the user and track their finger movements in real-time. The proposed method will involve recognizing the hand gestures, mapping them to specific commands, and sending those commands to the desktop application for execution. The project will involve developing a machine learning model to accurately recognize the gestures, as well as creating a software interface for mapping the gestures to application commands. The system will be tested on various applications such as media player, web browser, and document editor. The project's success will be measured by its efficiency, accuracy, and usability, with the goal of achieving an average recognition rate of 95%. The proposed system has the potential to provide an alternative input method for people with disabilities or anyone seeking a hands-free, intuitive method of controlling desktop applications. The project's outcomes will contribute to the development of new human-computer interaction techniques and have implications for various fields, including accessibility, gaming, and virtual reality.

Table of Content

Introduction

Cursor control of desktop applications refers to the ability to move the mouse pointer on the computer screen and interact with the graphical user interface (GUI) using various input devices. Coloured finger gestures is a method of cursor control that involves the use of coloured markers or gloves on the fingers to track movement and translate it into mouse pointer movement on the screen.

This method can be used with various software applications such as image editing, drawing programs, and video editing, where precise control of the mouse pointer is required. The coloured markers or gloves used in this technique are equipped with sensors that track the movement of the fingers and transmit the information to the computer. The software on the computer interprets this information and moves the mouse pointer accordingly.

The use of coloured finger gestures can be especially beneficial for people with disabilities, such as those who have limited mobility in their arms or hands. It can also be useful for individuals who have difficulty using traditional input devices such as a mouse or keyboard.

Overall, coloured finger gestures provide an innovative and intuitive way to control the cursor of desktop applications, offering a more natural and fluid interaction with the computer.

Scope

Hardware: The project may involve designing and building hardware components such as the coloured markers or gloves that will be used for tracking finger movements.

Software: The project may require developing software applications or modifying existing ones to interpret and respond to finger movements accurately.

User Interface Design: The project may involve designing a user interface that is optimised for the use of coloured finger gestures, with features such as customizable gesture recognition, feedback mechanisms, and accessibility options.

Testing and Validation: The project may involve testing and validating the hardware and software components to ensure that they work together seamlessly and provide the intended functionality.

User Experience: The project may focus on improving the user experience of cursor control for desktop applications using coloured finger gestures, including usability testing and feedback from users.

Integration with Existing Systems: The project may involve integrating the coloured finger gesture technology with existing systems and applications, such as assistive technologies or video editing software.

Motivation

Accessibility: Coloured finger gestures can provide an alternative input method for people with disabilities who have difficulty using traditional input devices such as a mouse or keyboard. This technology can offer greater accessibility and inclusion for individuals with limited mobility in their arms or hands.

Natural and Intuitive Interaction: Coloured finger gestures offer a more natural and intuitive way to interact with desktop applications, providing a more fluid and seamless experience. This technology can improve the user experience for individuals who prefer a more hands-on approach to computer interaction.

Precision and Control: Coloured finger gestures can provide a higher degree of precision and control over the mouse pointer, which can be beneficial for applications such as image editing, video editing, and drawing programs. This technology can improve the accuracy and efficiency of tasks that require precise mouse movement.

Innovation and Advancement: Coloured finger gestures represent an innovative and emerging technology that can drive advancements in human-computer interaction. This technology can push the boundaries of what is possible in terms of how we interact with computers and can pave the way for future advancements in this field.

Aim of the Project

The aim of a project using coloured finger gestures for cursor control of desktop applications is to create a technology that offers a more accessible, natural, and intuitive way for individuals to interact with their computer. The specific aim of the project may vary depending on the goals and objectives set by the project team, but here are some potential aims:

Develop a reliable and accurate system for tracking finger movements: The project may aim to create a system that accurately tracks the movement of the coloured markers or gloves worn on the fingers, with minimal lag or errors.

Create a customizable and adaptable system: The project may aim to create a system that allows users to customise their finger gestures and adapt the technology to their needs and preferences.

Improve the user experience of cursor control for desktop applications: The project may aim to create a system that offers a more natural and intuitive way to interact with desktop applications, improving the user experience and efficiency of tasks that require precise mouse movement.

Enhance accessibility for people with disabilities: The project may aim to create a system that offers an alternative input method for people with disabilities who have difficulty using traditional input devices such as a mouse or keyboard, improving their accessibility to computers.

Push the boundaries of human-computer interaction: The project may aim to create a technology that advances the field of human-computer interaction and offers new possibilities for how we interact with computers.

Overall, the aim of a project using coloured finger gestures for cursor control of desktop applications is to create a technology that offers a more accessible, natural, and intuitive way for individuals to interact with their computer, while also pushing the boundaries of human-computer interaction.

Objectives

The objectives of a proposed work using coloured finger gestures for cursor control of desktop applications will depend on the specific goals and scope of the project. However, here are some potential objectives:

Design and develop hardware components: The objective may be to design and develop coloured markers or gloves that can accurately track finger movements and transmit this information to the computer.

Develop software applications: The objective may be to develop software applications or modify existing ones to interpret and respond to finger movements accurately.

Create a user interface optimised for coloured finger gestures: The objective may be to design a user interface that is optimised for the use of coloured finger gestures, with features such as customizable gesture recognition, feedback mechanisms, and accessibility options.

Test and validate the technology: The objective may be to test and validate the hardware and software components to ensure that they work together seamlessly and provide the intended functionality.

Evaluate the user experience: The objective may be to evaluate the user experience of cursor control for desktop applications using coloured finger gestures, including usability testing and feedback from users.

Analyse the impact on accessibility: The objective may be to analyse the impact of coloured finger gestures on accessibility for people with disabilities who have difficulty using traditional input devices such as a mouse or keyboard.

Explore potential use cases and applications: The objective may be to explore potential use cases and applications of coloured finger gestures for cursor control in various industries, such as design, video editing, or assistive technologies.

Overall, the objectives of a proposed work using coloured finger gestures for cursor control of desktop applications can vary depending on the goals and scope of the project. The objectives may include hardware and software development, user interface design, testing and validation, user experience evaluation, accessibility analysis, and exploring potential use cases and applications.

Literature Survey

Author	**Year**	**Publisher**	**Title**	**Description**
Shah, K. N., Rathod, K. R., & Agravat, S. J.	2014	arXiv	A survey on human computer interaction mechanism using finger tracking.	This paper represents most of the innovative mechanisms of the finger tracking used to interact with a computer system using computer vision. Users can interact with a computer system without using any conventional peripheral devices. Marker is used to recognize hand movement accurately & successfully. Numbers of applications for the finger tracking are developed by researchers. They always try to minimise the number of peripheral devices to interact with the computer. So, it is more useful for the physically disabled people.
Xu, P.	2017	arXiv	A real-me hand gesture recognition and co-human interactions	This project is about designing a real-time human-computer interaction system based on hand gestures. The whole system consists of three components: hand detection, gesture recognition and human-computer interaction (HCI) based on recognition; and realises the robust control of mouse and keyboard events with a higher accuracy of gesture recognition.

				Specifically, we use the convolutional neural network (CNN) to recognize gestures and make it attainable to identify relatively complex gestures using only one cheap monocular camera.
Patil, A. M., Dudhane, S. U Gandhi, M. B., Uke, N. J.	2013	IZARC	Cursor Control System Using Gesture Recognition	This paper presents our on-going development of machine- user interface which implements hand gesture recognition using simple computer vision and multimedia techniques. The interface allows using their hands to control the cursor as well as easily move and resize windows that are open on their screen. Our aim is to enable humans to interface with the machine and interact naturally without any mechanical devices. Anyone with a computer and a camera should be able to take full advantage of this project. First the input image is converted to a binary image to separate the hand from the background. Then the centre of the hand is calculated and the calculated radius of the hand is found. Fingertip points are calculated using the Convex Hull algorithm. All the mouse movements are controlled using the hand gesture.
Dhote, A. P., Parihar, V. R., Sonar , R. P., & Nage, R. S.	2017	IEEE	A novel system of real time hand tracking and gesture recognition	A novel system of real time hand tracking and gesture recognition

Banerjee, A., Ghosh, A., Bharadwaj, K., & Saikia, H	2014	arXiv	Mouse control using a web camera based on colour detection.	In this paper they present an approach for Human computer Interaction (HCI), where we have tried to control the mouse cursor movement and click events of the mouse using hand gestures. Hand gestures were acquired using a camera based on colour detection technique. This method mainly focuses on the use of a Web Camera to develop a virtual human computer interaction device in a cost-effective manner.
Kumar, V., Mahe, S., & Vyawahare, S.	2016	JNCET	Mouse simulation using two coloured tapes.	In this paper, they present a novel approach for Human Computer Interaction (HCI) where we control cursor movement using a real-time camera. Current methods involve changing mouse parts such as adding more buttons or changing the position of the tracking ball. Instead, our method is to use a camera and computer vision technology, such as image segmentation and gesture recognition, to control mouse tasks and we show how it can perform everything as current mouse devices can. The software can be used as an intuitive input interface to applications that require multidimensional control e.g., computer games etc.
Grif, H. S., & Farcas, C. C.	2016	-	Mouse Cursor Control System Based on Hand	This paper presents a HCI interface for mouse cursor control. The purpose of the

			Gesture.	implemented solution is to control the mouse cursor by user hand gestures captured through a webcam. For improving the gesture recognition based on the fluctuation of illuminance levels the finger strips colour detection was used. The results reveal the good behaviour of the system in low light conditions. application has an advantage in using colour detection for gesture interpretation, because it can be used in low or high intensity light with more precision than other applications and in the same time, it opens a window for further applications that intend to control any device using hand gestures or body movements. The disadvantage of this type of application is that the final result is influenced by the quality and fps of the chosen webcam.

Related Work

Finger Shadow: A System for Finger-Mounted Marker Tracking and Gesture Recognition: This paper describes a system for finger-mounted marker tracking and gesture recognition that enables users to control a computer cursor using coloured finger markers.

Fingertip Interaction Techniques for Wearable AR/VR: This paper describes various fingertip interaction techniques for wearable AR/VR systems, including finger tracking and gesture recognition, that enable users to interact with virtual objects and control the system.

Leap Motion: This is a commercial product that uses infrared cameras and software to track hand and finger movements, enabling users to control their computers without a mouse or keyboard.

Microsoft Kinect: This is another commercial product that uses depth sensors and cameras to track body and hand movements, enabling users to control their computers and interact with virtual objects.

Accessible Interfaces for Motor-impaired Users: This paper describes various accessible interfaces, including finger and hand tracking, that enable motor-impaired users to control their computers and perform tasks that would otherwise be difficult or impossible.

Assistive Technology for People with Disabilities: This book provides an overview of various assistive technologies, including input devices and software, that enable people with disabilities to access computers and perform tasks independently.

Requirement Analysis

Hardware Requirements: The project may require the development or acquisition of hardware components, such as coloured markers or gloves, that can accurately track finger movements and transmit this information to the computer.

Software Requirements: The project may require the development or modification of software applications that can interpret and respond to finger movements accurately. This may include programming languages, software frameworks, and libraries.

User Interface Requirements: The project may require the design of a user interface that is optimised for the use of coloured finger gestures, with features such as customizable gesture recognition, feedback mechanisms, and accessibility options.

Performance Requirements: The project may require the hardware and software components to meet certain performance requirements, such as speed, accuracy, and reliability.

Compatibility Requirements: The project may require the hardware and software components to be compatible with various desktop applications and operating systems.

Usability Requirements: The project may require the hardware and software components to be easy to use and learn, with minimal training or user manuals required.

Accessibility Requirements: The project may require the hardware and software components to be accessible to people with disabilities, such as those with motor impairments, visual impairments, or hearing impairments.

Testing and Validation Requirements: The project may require the testing and validation of the hardware and software components to ensure that they work together seamlessly and provide the intended functionality.

Security Requirements: The project may require the hardware and software components to meet certain security requirements, such as data encryption and user authentication.

Stakeholder Identification

Users: The primary stakeholders are the users who will be using the system to control the cursor of their desktop applications. They will have a direct impact on the success of the project, and their feedback will be important for evaluating and refining the system.

Developers: The developers who will be designing and building the hardware and software components of the system are also key stakeholders. They will have a direct impact on the functionality and performance of the system, and their expertise will be essential for ensuring that the project meets its objectives.

Management: Management is another important stakeholder, as they will be responsible for overseeing the project, providing resources, and ensuring that the project aligns with the overall goals and objectives of the organisation.

Regulators: Regulatory bodies, such as government agencies or industry associations, may also be stakeholders in the project. They may have requirements or guidelines that the project must meet to ensure compliance with relevant laws and regulations.

Partners and Suppliers: Partners and suppliers who provide the hardware, software, or other resources necessary for the project may also be stakeholders. Their input and feedback will be important for ensuring that the project is delivered on time and within budget.

Investors: Investors who have provided funding or resources for the project may also be stakeholders. They will be interested in the project's success and may provide input and feedback on the project's progress.

Functional Requirements

Gesture Recognition: The system should be able to accurately recognize and interpret different finger gestures made with coloured markers or gloves.
Cursor Control: The system should be able to control the cursor on the desktop screen based on the finger gestures made by the user.
Click and Drag: The system should be able to recognize click and drag gestures, allowing the user to interact with desktop applications in a similar way to a traditional mouse.
Customizable Gestures: The system should allow users to customise and define their own finger gestures, providing a personalised and flexible user experience.
Sensitivity Adjustment: The system should allow users to adjust the sensitivity of the finger gestures, allowing for a more personalised and comfortable experience.
Compatibility: The system should be compatible with different operating systems and hardware configurations, allowing for a wide range of users to benefit from the system.

Non-Functional Requirements

Performance: The system should provide fast and responsive cursor control, with minimal delay or lag between the user's finger gestures and the movement of the cursor on the screen.
Reliability: The system should be reliable and robust, with minimal downtime or system failures. The system should also be able to recover gracefully from errors or failures.
Usability: The system should be easy to use and intuitive, with a simple and clear user interface. The system should also provide clear feedback to the user to confirm their finger gestures have been recognized and to indicate the current mode of operation (e.g., click or drag).
Security: The system should be secure, protecting user data and information from unauthorised access or tampering.
Compatibility: The system should be compatible with different hardware and software configurations, allowing for a wide range of users to benefit from the system.
Accessibility: The system should be accessible to users with disabilities, with features such as magnification, voice control, or other assistive technologies.

System Requirements

Hardware: The system may require a computer with a camera or webcam capable of capturing video footage of the user's-coloured finger gestures. The camera should be of sufficient resolution to accurately capture and interpret the gestures.
Software: The system may require specialised software to recognize and interpret the finger gestures made by the user. This software should be compatible with the operating system being used by the user, such as Windows, macOS, or Linux.
Coloured Markers or Gloves: The user may need to use coloured markers or gloves to make the finger gestures recognizable to the system. The markers or gloves should be easily accessible and inexpensive.
Internet Connection: Depending on the design of the system, an internet connection may be required to support cloud-based processing, machine learning, or other advanced features.
User Interface: The system should provide a user-friendly interface for controlling the cursor, with clear visual feedback to indicate the current mode of operation and the gestures being made by the user.
Development Tools: The system may require specialised development tools to build and test the software, such as an integrated development environment (IDE), code editors, or testing frameworks.

Hardware Requirements

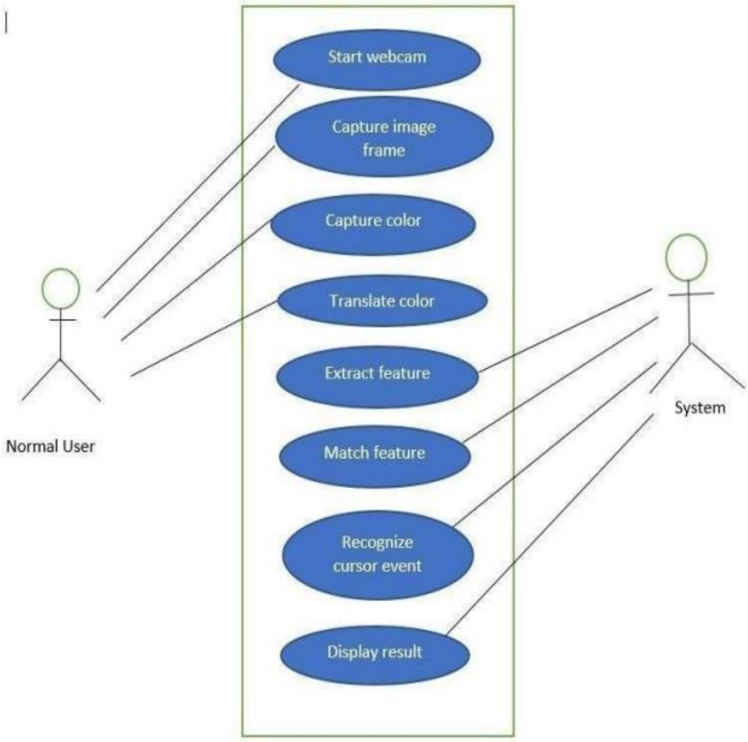
Camera: The system may require a camera or webcam capable of capturing video footage of the user's-coloured finger gestures. The camera should be of sufficient resolution to accurately capture and interpret the gestures. A camera with a resolution of at least 720p or higher is recommended.
Processor: The system may require a processor with sufficient processing power to interpret the coloured finger gestures in real-time. A quad-core processor with a clock speed of at least 2 GHz or higher is recommended.
Memory: The system may require sufficient memory to store the software and interpret the coloured finger gestures. A minimum of 4 GB of RAM is recommended.
Graphics card: The system may require a graphics card to support real-time video processing and display. A graphics card with at least 1 GB of video memory is recommended.

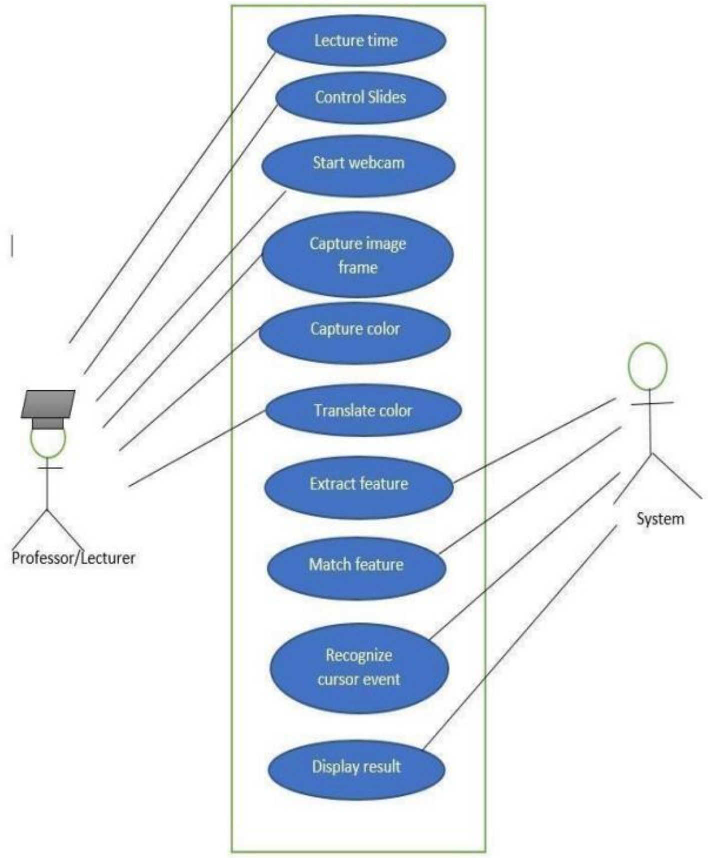
Software Requirements

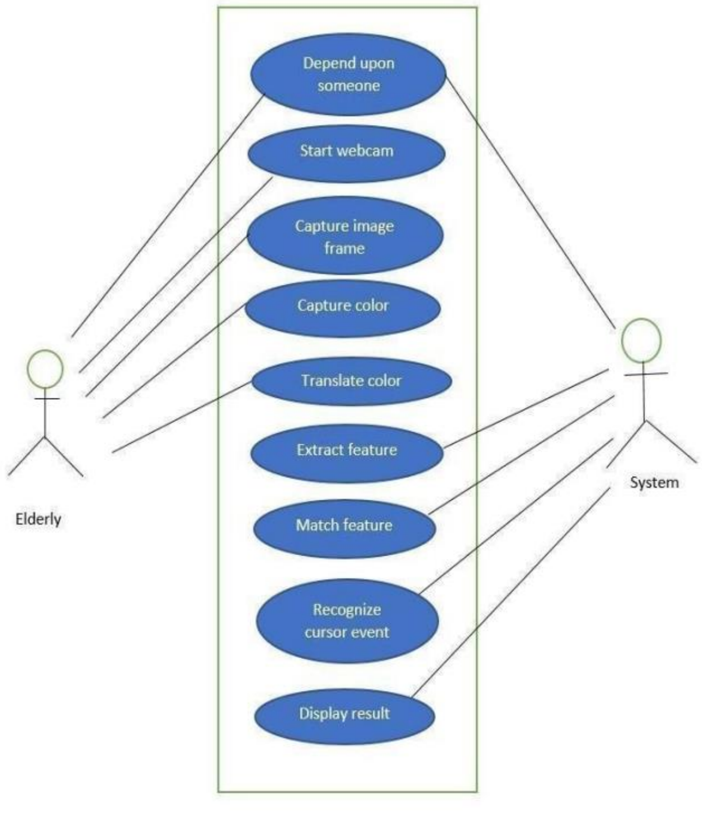
MATLAB: The project will require the latest version of MATLAB software installed on the computer.
Computer Vision Toolbox: This toolbox is needed to work with the video and image processing operations required for gesture recognition and cursor control.
Image Processing Toolbox: This toolbox is required for pre-processing and feature extraction from the images or video frames.
Compiler: A MATLAB Compiler can be used to generate an executable file for the system, which can be used on other computers without MATLAB installed.

Diagrams

Use Case Diagram




Lecture time
Control Slides
Start webcam
Capture image frame
Capture color
Translate color
Extract feature
Match feature
Recognize cursor event
Display result
Professor/Lecturer
System


Depend upon someone
Start webcam
Capture image frame
Capture color
Translate color
Extract feature
Match feature
Recognize cursor event
Display result
Elderly
System

Proposed Methodology

The proposed methodology is a computer vision-based approach for mouse control using a webcam or other video input device. The approach relies on detecting colours in the video stream to control the mouse pointer, simulate mouse clicks, and scroll events.

The system uses a thresholding technique to detect three different colours: red, green, and blue. The threshold values for each colour can be adjusted using the input parameters to the function. The red colour is used to detect the position of the mouse pointer, while the blue colour is used to simulate left-click, right-click, and double-click events. The green colour is used to simulate mouse scrolling events.

The system uses a blob analysis algorithm to detect the centroid and bounding boxes of objects in the video stream that match the specified colour thresholds. Once the objects are detected, the system applies different routines depending on the number of detected objects and their position in the video stream. For example, if only one red object is detected, the system moves the mouse pointer to its centroid position. If one blue object is detected, the system simulates a left-click event. If two blue objects are detected, the system simulates a right-click event, and if three blue objects are detected, the system simulates a double-click event. If green objects are detected, the system simulates a scroll event depending on the position of the centroids of the green objects relative to the previous frame.

The system uses a JAVA robot object to simulate mouse and keyboard events. The system includes a graphical user interface (GUI) that displays the output video stream and allows the user to adjust the colour threshold values in real-time.

Overall, the proposed methodology provides a simple and low-cost solution for mouse control using a webcam or other video input device, which can be useful for applications that require hands-free or remote control of the computer mouse.

Implementation with Code

Code:

```
function MouseControl(redThresh, greenThresh, blueThresh, numFrame)
warning('off','vision:transition:usesOldCoordinates');

%% Initialization`
if nargin < 1
    redThresh = 0.22;  % Threshold for Red color detection
    greenThresh = 0.14; % Threshold for green color detection
    blueThresh = 0.18; % Threshold for blue color detection
    numFrame = 900; % Total number of frames duration
end

cam = imaqhwinfo; % Get Camera information
cameraName = char(cam.InstalledAdaptors(end));
cameraInfo = imaqhwinfo(cameraName);
cameraId = cameraInfo.DeviceInfo.DeviceID(end);
cameraFormat = char(cameraInfo.DeviceInfo.SupportedFormats(end));

jRobot = java.awt.Robot; % Initialize the JAVA robot
vidDevice = imaq.VideoDevice(cameraName, cameraId, cameraFormat, ...
% Input Video from current adapter
                    'ReturnedColorSpace', 'RGB');

vidInfo = imaqhwinfo(vidDevice);  % Acquire video information
screenSize = get(0,'ScreenSize'); % Acquire system screensize
hblob = vision.BlobAnalysis('AreaOutputPort', false, ... % Setup
blob analysis handling
                                'CentroidOutputPort', true, ...
                                'BoundingBoxOutputPort', true', ...
                                'MaximumBlobArea', 3000, ...
                                'MinimumBlobArea', 100, ...
                                'MaximumCount', 3);
hshapeinsBox = vision.ShapeInserter('BorderColorSource', 'Input
port', ... % Setup colored box handling
                                    'Fill', true, ...
                                    'FillColorSource', 'Input port',
...
                                    'Opacity', 0.4);
hVideoIn = vision.VideoPlayer('Name', 'Final Video', ... % Setup
output video stream handling
                                'Position', [100 100
vidInfo.MaxWidth+20 vidInfo.MaxHeight+30]);
nFrame = 0; % Initializing variables
lCount = 0; rCount = 0; dCount = 0;
sureEvent = 5;
```

```
iPos = vidInfo.MaxWidth/2;

%% Frame Processing Loop
while (nFrame < numFrame)
    rgbFrame = step(vidDevice); % Acquire single frame
    rgbFrame = flipdim(rgbFrame,2); % Flip the frame for
userfriendliness
    diffFrameRed = imsubtract(rgbFrame(:,:,1), rgb2gray(rgbFrame));
% Get red components of the image
    binFrameRed = im2bw(diffFrameRed, redThresh); % Convert the
image into binary image with the red objects as white
    [centroidRed, bboxRed] = step(hblob, binFrameRed); % Get the
centroids and bounding boxes of the red blobs

    diffFrameGreen = imsubtract(rgbFrame(:,:,2),
rgb2gray(rgbFrame)); % Get green components of the image
    binFrameGreen = im2bw(diffFrameGreen, greenThresh); % Convert
the image into binary image with the green objects as white
    [centroidGreen, bboxGreen] = step(hblob, binFrameGreen); % Get
the centroids and bounding boxes of the blue blobs

    diffFrameBlue = imsubtract(rgbFrame(:,:,3), rgb2gray(rgbFrame));
% Get blue components of the image
    binFrameBlue = im2bw(diffFrameBlue, blueThresh); % Convert the
image into binary image with the blue objects as white
    [~, bboxBlue] = step(hblob, binFrameBlue); % Get the centroids
and bounding boxes of the blue blobs

    if length(bboxRed(:,1)) == 1 % Mouse pointer movement routine

jRobot.mouseMove(1.5*centroidRed(:,1)*screenSize(3)/vidInfo.MaxWidth
, 1.5*centroidRed(:,2)*screenSize(4)/vidInfo.MaxHeight);
    end
    if ~isempty(bboxBlue(:,1)) % Left Click, Right Click, Double
Click routine
        if length(bboxBlue(:,1)) == 1 % Left Click routine
            lCount = lCount + 1;
            if lCount == sureEvent % Make sure of the left click
event
                jRobot.mousePress(16);
                pause(0.1);
                jRobot.mouseRelease(16);
            end
        elseif length(bboxBlue(:,1)) == 2 % Right Click routine
            rCount = rCount + 1;
            if rCount == sureEvent % Make sure of the right click
event
                jRobot.mousePress(4);
```

```
                pause(0.1);
                jRobot.mouseRelease(4);
            end
        elseif length(bboxBlue(:,1)) == 3 % Double Click routine
            dCount = dCount + 1;
            if dCount == sureEvent % Make sure of the double click
event
                jRobot.mousePress(16);
                pause(0.1);
                jRobot.mouseRelease(16);
                pause(0.2);
                jRobot.mousePress(16);
                pause(0.1);
                jRobot.mouseRelease(16);
            end
        end
    else
        lCount = 0; rCount = 0; dCount = 0; % Reset the sureEvent
counter
    end
    if ~isempty(bboxGreen(:,1)) % Scroll event routine
        if (mean(centroidGreen(:,2)) - iPos) < -2
            jRobot.mouseWheel(-1);
        elseif (mean(centroidGreen(:,2)) - iPos) > 2
            jRobot.mouseWheel(1);
        end
        iPos = mean(centroidGreen(:,2));
    end
    vidIn = step(hshapeinsBox, rgbFrame, bboxRed,single([1 0 0])); %
Show the red objects in output stream
    vidIn = step(hshapeinsBox, vidIn, bboxGreen,single([0 1 0])); %
Show the green objects in output stream
    vidIn = step(hshapeinsBox, vidIn, bboxBlue,single([0 0 1])); %
Show the blue objects in output stream
    step(hVideoIn, vidIn); % Output video stream
    nFrame = nFrame+1;
end
%% Clearing Memory
release(hVideoIn); % Release all memory and buffer used
release(vidDevice);
clc;
end
```

Output:

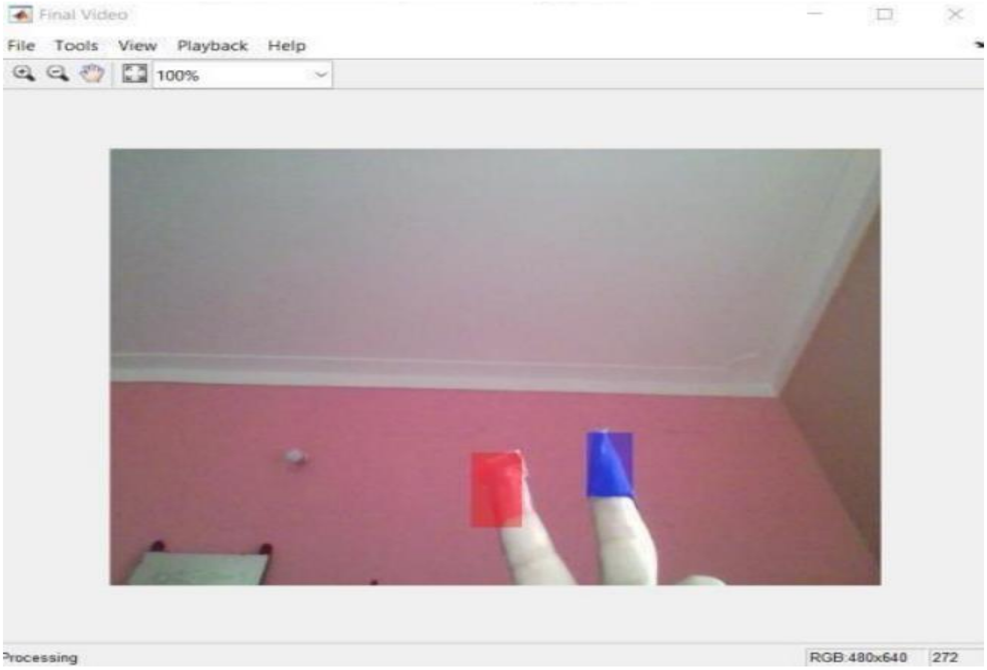


Figure: Detection of colour and performing left click action

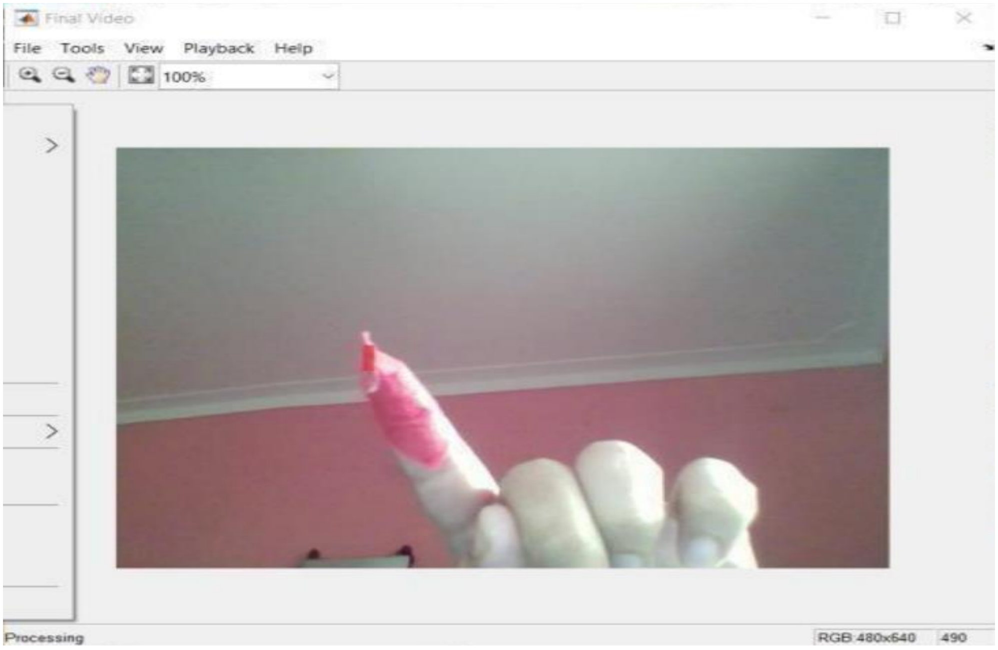


Fig: Detection of red colour to hover through the menu

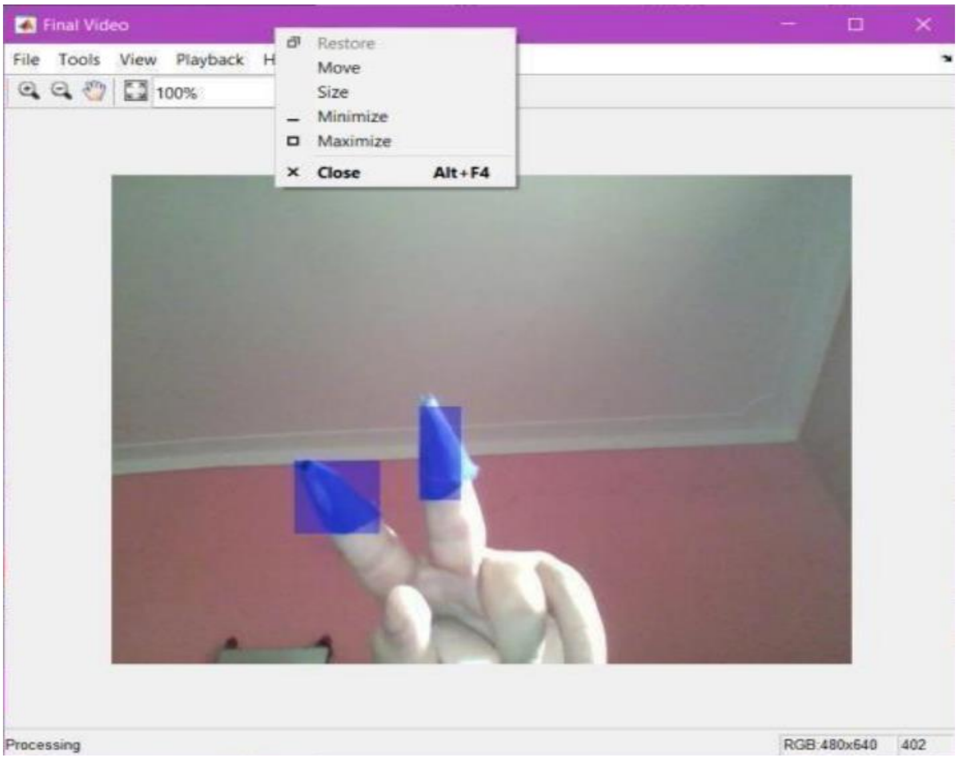


Figure: Using 2 blue fingers to perform right click action

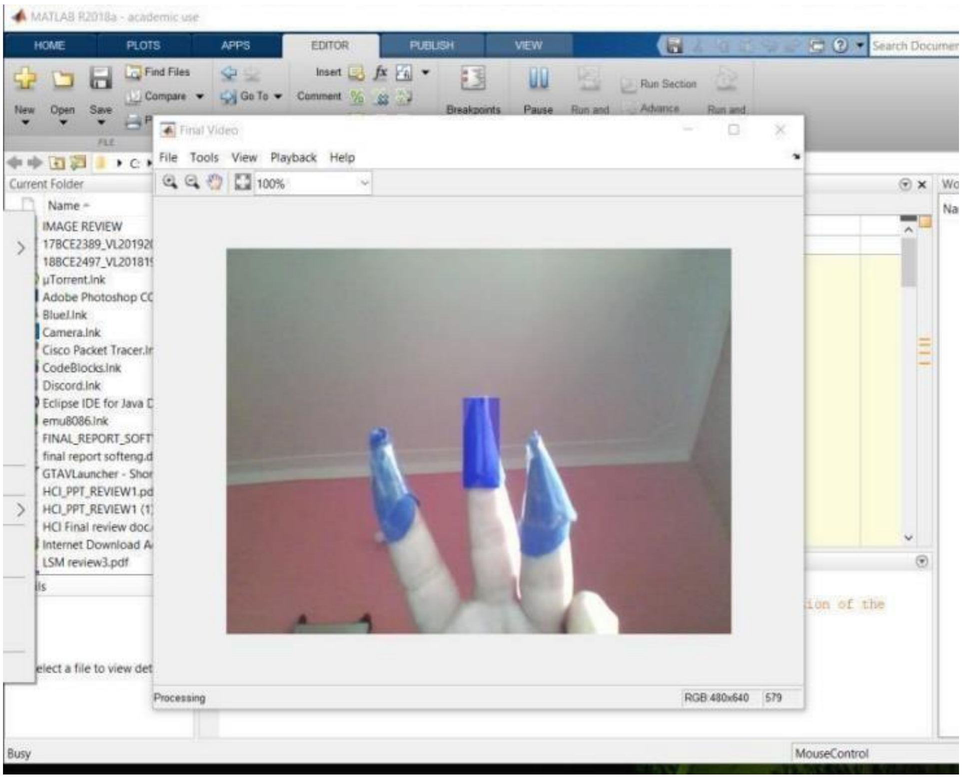


Figure: Using 3 blue fingers to perform double click action

Results and Discussion

Moving slides in ppt: PASS

Method used: 1 red finger and 1 blue finger to scroll cursor to required position

Highlighting text in ppt: PASS

Method used: 1 red finger and 1 blue finger dragging to required position

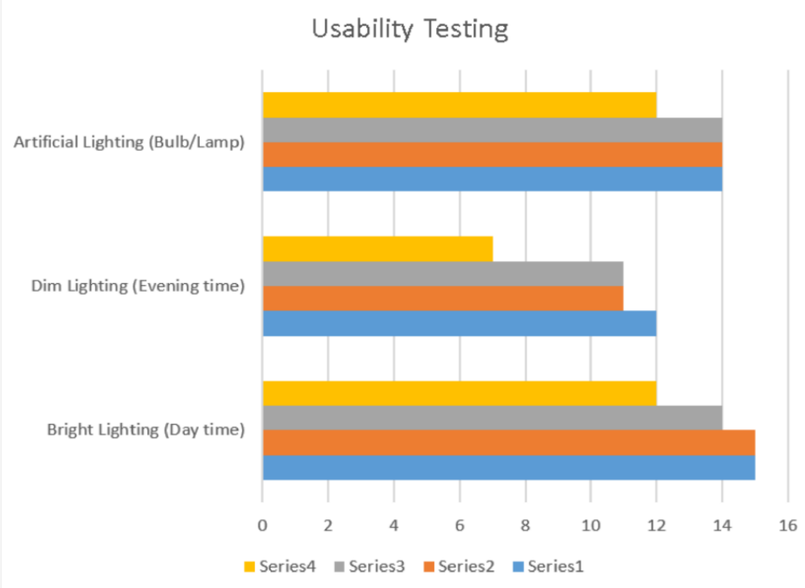
Selecting presentation file in proper light: PASS

Method used: Selected presentation file from desktop directory

Selecting presentation file in dim light: FAIL

Method used: Selected presentation file from desktop directory

Reason for failure: Poor colour detection in dim light



Summary of Evaluations:

Using the Model Human Processor, the response time for the task was calculated as 800 msec. The perceptual processor remained constant throughout the task, while the cognitive and motor processors had processing times of 490 msec and 210 msec, respectively.

Using the Keystroke Level Model and applying the M placement heuristics, the total execution time for the task was calculated as 2.65 seconds. The task involved raising 1 red finger, pointing to the required file, and raising 1 blue finger.

Conclusion and Future Work

Based on the results of our project, we can conclude that hand gesture recognition can be a viable alternative to traditional mouse control, as it allows for simple operations such as clicking, scrolling, and hovering to be performed accurately using hand gestures. However, the system has limitations, such as the requirement for a well-lit room and the potential difficulty in handling the cursor in certain situations.

Test results showed that the system had an accuracy rate of over 88-91% in well-lit bright lighting conditions but dropped to about 70% in dim lighting conditions. The accuracy rate could be improved with a better-quality camera. The project team conducted a survey and found out that previous similar projects that used hand gestures to control the mouse cursor had an accuracy rate of not more than 65-70%.

Future work on this project could include improving the accuracy of the system by using a better-quality camera or improving the image processing algorithms. Additionally, further research could be conducted to investigate the feasibility of using hand gesture recognition for more complex tasks, such as typing or navigating a computer interface. Overall, this project demonstrates the potential for hand gesture recognition to be a valuable tool in computer interaction, and highlights areas for further development and exploration.

References

1. Shah, K. N., Rathod, K. R., & Agravat, S. J. (2014). A survey on human computer interaction mechanism using finger tracking. arXiv preprint arXiv:1402.0693.
2. Xu, P. (2017). A real-time hand gesture recognition and human-computer interaction system. arXiv preprint arXiv:1704.07296.
3. Grif, H. S., & Farcas, C. C. (2016). Mouse Cursor Control System Based on Hand Gesture. Procedia Technology, 22, 657-661.
4. Dhote, A. P., Parihar, V. R., Sonar, R. P., & Nage, R. S. (2017, November). A novel system of real time hand tracking and gesture recognition. In 2017 International Conference on Inventive Computing and Informatics (ICICI) (pp. 512-516). IEEE.
5. Patil, A. M., Dudhane, S. U., Gandhi, M. B., & Uke, N. J. (2013). Cursor Control System Using Hand Gesture Recognition. International Journal of Advanced Research in Computer and Communication Engineering.
6. Banerjee, A., Ghosh, A., Bharadwaj, K., & Saikia, H. (2014). Mouse control using a web camera based on colour detection. arXiv preprint arXiv:1403.4722.
7. Niyazi, K., Kumar, V., Mahe, S., & Vyawahare, S. (2016). Mouse simulation using two coloured tapes. Department of Computer Science, University of Pune, India, International Journal of Information Sciences and Techniques (IJIST) Vol, 2.
8. Oliveira Neto F. M., Han L. D, G., Jeong M. K., 2013, “An Online Self Learning Algorithm For Gesture Recognition”, IEEE Transactions on Intelligent Transportation Systems, Volume 5, Issue 4.
9. Kate R., 2013, “Color Recognition Using Segmentation”, International Journal of Engineering Research and Technology (IJERT), Vol. 1, No. 9.
10. Du S., Ibrahim M., Shehata M. and Badawy W., 2013, “Image Recognition: A State-of-the-Art Review”, IEEE Transactions on Circuits and Systems For Video Technology, Vol. 23, No. 2, pp. 311-325.

Link