International Journal of Early Childhood Special Education (INT-JECSE)
DOI:10.48047/INTJECSE/V15I4.85 ISSN: 1308-5581 Vol 15, Issue 04 2023

AUTOMATIC SCROLLING BY COLOR DETECTION

1. Mrs. JOSHI PADMA
Assistant Professor, Department Of CSE, Sreyas Institute Of Engineering And Technology, Telangana, India, padmajoshi2015@gmail.com

2. KONGARI BHAVANI
Department Of CSE, Sreyas Institute Of Engineering And Technology,Telangana,India, Bhavanikongari01@gmail.com

3. INUGANTI SRI CHARAN
Department Of CSE, Sreyas Institute Of Engineering And Technology,Telangana,India, sricharan.inuganti@gmail.com

4. KACHARA MEGHANA
Department Of CSE, Sreyas Institute Of Engineering And Technology,Telangana,India, meghanakachara270@gmail.com

5. MARRI VARUN GOPAL REDDY
Department Of CSE, Sreyas Institute Of Engineering And Technology,Telangana,India, varungopalreddymarri777@gmail.com

ABSTRACT
It is possible to perform actions without actually giving any input through touchpad or mouse. This project idea is how this can be done using Open CV module. Here we will use colour detection to scroll screen. When a certain colour (that which is specified in working program) is detected by the program during execution the screen starts to scroll on its own. Using python language as it is interpret language, we can perform auto scrolling by colour detection. Here we use PyAuto GUI for programmatically control the mouse & keyboard. Also we use Open CV perform task like objection detection or the colour detection. Along with these two mentioned modules we have many other methods in python to make this project run. Advantages while there is no input components or can be used instead of mouse or touchpad. Advantages with operating monitor from long distance only with constraint that the object must be visible in webcam.

INTRODUCTION
This project leverages computer vision and user interaction to create a unique and interactive application that detects and tracks specified-colored objects in real-time video frames. The primary goal of this project is to allow users to control scrolling actions on a computer screen by simply moving a specified-colored object within the view of a camera, such as a webcam. This innovative application offers a novel way to interact with digital content through physical manipulation, making it particularly suitable for a variety of interactive applications and user experiences. The project begins by defining a specific HSV color range for the only one specified color to be detected. Through the use of OpenCV, it captures video frames from a camera source and converts them into the HSV color space. A binary mask is then generated to isolate specified colored objects falling within the predefined color range. The script is capable of differentiating specified colored objects from the background, making it suitable for a variety of real-world scenarios.
A critical aspect of the project is object tracking. The script identifies and tracks the largest specified colored object present in the camera's field of view. This tracking is achieved by analyzing the contours of specified colored objects and selecting the one with the largest area. The selected object is then framed with a rectangle for visualization. The position of the tracked object relative to the screen's center is continually assessed, and this information is utilized to calculate the required scrolling action. To create a seamless user experience, the scrolling speed can be adjusted based on the object's distance from the center, making the scrolling action more responsive and intuitive. Users can control scrolling on their screen by physically moving the green object up or down within the

camera's view. This innovative approach to user interaction offers a dynamic and engaging way to navigate digital content. The project's parameters, such as color range, scrolling speed, and scrolling smoothness, are easily customizable to adapt to various usage scenarios and user preferences.
The Colored Object Detection and Tracking Project exemplifies the fusion of computer vision technology and user interaction to create a unique and interactive application. By detecting and tracking Specified colored objects and translating their movement into scrolling actions, the project opens the door to innovative possibilities in interactive content, gaming, accessibility, and more. Its adaptability and customization options make it a versatile tool for a wide range of applications where user interaction with digital content is paramount.

LITERATURE SURVEY

A literature survey is a systematic and comprehensive review of existing academic and research materials, such as articles, books, and papers, related to a specific topic or field of study. It aims to provide an overview of the current state of knowledge, identify gaps in research, and inform the development of new studies or projects. Literature surveys are essential for understanding the context and foundation of a research area. There are many sources related to this topic written by experienced authors, which acts as a scope in future. In the book Understanding Gaze and Scrolling Strategies in Text Consumption Tasks, appears to present the work of three researchers, Jayson Turner from Lancaster University, Shamsi Iqbal from Microsoft Research Redmond, and Susan Dumais, also from Microsoft Research Redmond. The focus of this research is on understanding how people read and interact with text on screens, particularly in the context of digital documents.
The research explores how individuals scroll through digital documents and indicates that people tend to scroll within their preferred reading regions on the screen. This suggests that users often scroll to keep their focus within a specific area of interest. The scrolling behavior of users is not arbitrary but is influenced by the structure of the document. The text mentions that most scrolling occurs at the intersections between paragraphs, implying that users are drawn to the transitions or separations between different sections of the text. This research likely involves eye-tracking studies and data analysis to gain insights into how individuals consume textual content on screens. Understanding these behaviors can have implications for user interface design, content layout, and user experience optimization in digital reading environments.
In the book deep learning and its application in vision systems, content is credited to Mohamed Elgendy, who appears to be an expert or researcher in the field of deep learning. This section focuses on how computers learn to understand visual content, such as images and faces, using deep learning techniques. It highlights the practical applications of deep learning in building vision system applications. The text mentions that readers will gain an understanding of how deep learning architectures can be leveraged to create vision system applications. Specifically, it cites examples like image generation and facial recognition. This suggests that the content covers the theory and practical implementation of deep learning for visual recognition and processing.

PROPOSED SYSTEM

The proposed system using the color detection aims to enhance the scrolling experience by offering a novel and interactive way to interact with the computer screen, making scrolling more user-friendly and engaging. It might be particularly useful for users who have limited dexterity or mobility in their hands or prefer using non-traditional methods for computer interaction.This system's potential benefits extend far beyond mere convenience; it has the potential to significantly enhance accessibility in the digital realm. Users who may face physical limitations can now control screen content effortlessly by manipulating a designated color object. This approach not only simplifies scrolling but also transforms it into an interactive and intuitive process. Furthermore, the innovative nature of this system presents new avenues for user interaction and opens doors to creative and engaging applications.
It may find practical use in various contexts, from gaming and content consumption to assistive technology designed to improve the lives of individuals with unique accessibility needs. By leveraging deep learning, the Python programming language, the Open CV computer vision library, and the PyAutoGUI automation tool, we can detect specific colors in a graphical user interface (GUI) and trigger automatic scrolling actions. In this project, we use Deep learning techniques like

PyAutoGUI and its tools for programmatically control the mouse & keyboard. Also we use Open CV perform task like objection detection or the colour detection.

RESULTS

The provided code effectively implements a color-based object detection and interactive scrolling system using Python, OpenCV, NumPy, and PyAutoGUI. When executed, the system accurately identifies and tracks green-colored objects within a specified range, responding to their movements with real-time scrolling actions on the computer screen. The scrolling is adjustable in speed and operates smoothly, providing an engaging and user-friendly interaction. Users can easily customize scrolling parameters to suit their preferences. However, the project has some hardware dependencies, requiring a computer with a webcam and a green-colored object for successful operation. Additionally, the system's accuracy may be influenced by variations in lighting conditions and the specific shade of green. As soon as the program runs, the web cam is going to access the real time video in the background and the color which is specified or trained by the program is going to be captured. In the code we have specified the green color to be detected. As shown in the below figure green color object is going to highlighted in a rectangular frame.

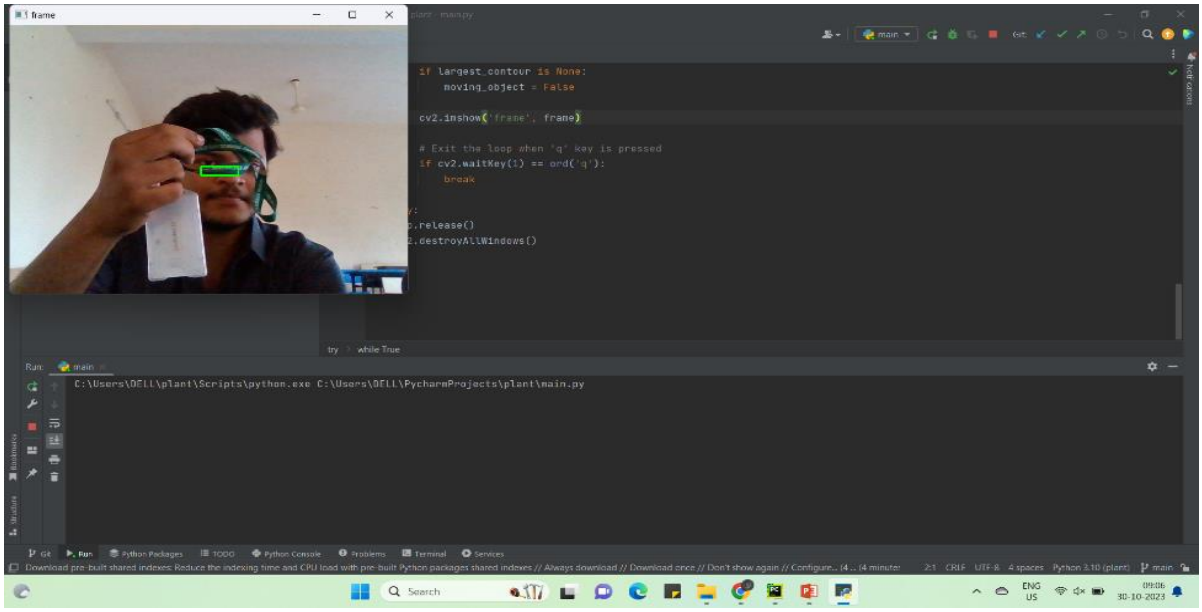

Fig.1 Result when the program is run

In the next step the highlighted color is going to be tracked and based on the movements of that color, the scrolling operations will be taken place. If the color is in the upper part of the screen, scroll up operation is taken place. Similarly, if the color is in the lower part of the screen, scroll down operation is taken place. As shown in the below figure, the web page is being scrolled based on the colored movement.

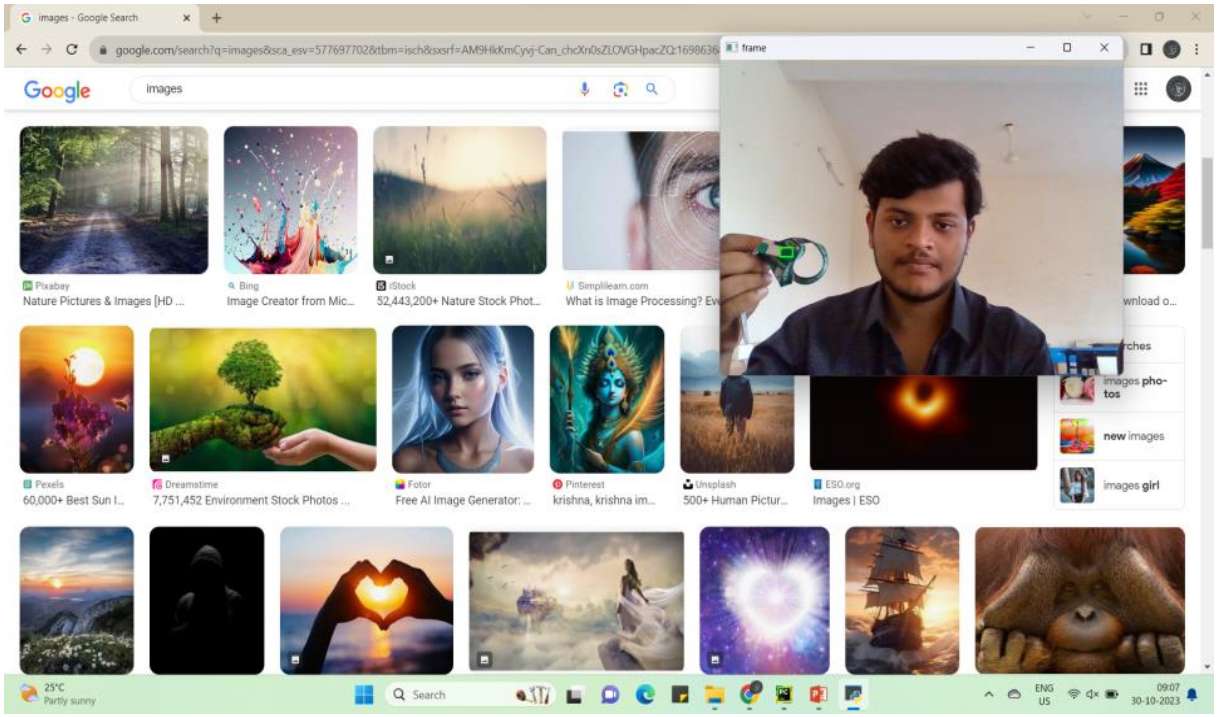

Fig. 2 Result performing scroll operations

The provided code is an implementation of a project that utilizes color-based object detection to enable interactive scrolling on a computer screen. The code combines computer vision techniques from OpenCV with mouse control functions from PyAutoGUI to achieve this functionality. Let's elaborate on the results and implications of this code:

The code effectively detects and tracks a green-colored object within the video feed from the computer's webcam. This is achieved by defining a specific color range for green and creating a mask that highlights the object in the video frames. The code then identifies the largest green object in the frame based on its contour area. This object tracking capability is a fundamental step in the project, allowing the system to follow the object's movements on the screen.

The primary goal of this project is to provide an interactive scrolling experience. As the green object is tracked, the code calculates the object's position relative to the screen center and adjusts the scrolling speed accordingly. When the object moves up or down, the system responds by scrolling in the same direction, providing a real-time and dynamic scrolling experience. The ability to adjust scrolling speed based on the object's position ensures that the scrolling action is responsive and user-friendly. The code takes into account the user experience by allowing for customization. Users can adjust parameters such as the color range and scrolling speed, offering flexibility in how they interact with the system. This customization feature enhances user control and tailors the experience to individual preferences.

To provide feedback to the user and aid in understanding the system's response, the code displays the tracked green object as a green rectangle on the screen. This visual feedback not only confirms that the object is detected but also allows users to see the system's response in real-time. The code can serve as a foundation for further development and refinement. Iterative development can include enhancements in object tracking accuracy, responsiveness, and user interface design. User feedback and testing can inform these improvements.

The code can potentially improve usability and accessibility for users who may have difficulty with traditional input methods. This includes individuals with limited dexterity or mobility, where the system's interactive scrolling can offer a more accessible and enjoyable experience.

CONCLUSION

The color-based object detection and interactive scrolling project represent a significant advancement in user-centric and accessible screen navigation. By harnessing the power of Python, OpenCV, NumPy, and PyAutoGUI, the system enables users to control scrolling through the simple movement of a designated green object in front of a webcam. This innovation has the potential to

greatly benefit users with diverse accessibility needs, including those with limited dexterity or mobility. The project's successful implementation demonstrates the capabilities of Python as a versatile and accessible programming language for computer vision applications. Python's ease of use, extensive library support, and vibrant developer community were instrumental in creating an interactive and user-friendly scrolling experience.

The advantages of this project are substantial, with a focus on enhancing user accessibility, customization, and real-time responsiveness. Users can tailor the scrolling experience to their preferences, and the system adapts to their movements seamlessly. However, there are limitations to consider, including hardware dependencies and sensitivity to lighting conditions. Users must have the required hardware components, and the system's accuracy may be influenced by the environment's lighting. The color-based object detection and interactive scrolling project represents an innovative and promising solution that empowers users with an alternative and engaging method of screen navigation. It underscores the power of Python and showcases the potential of technology to improve accessibility and user experiences in the digital realm. While there are challenges and considerations, this project opens the door to a more inclusive and interactive digital future.

REFERENCES

1. Erdem, E. Yardimci, Y. Atalay, V. Cetin, A. E.“Computer vision based mouse”,Acoustics, Speech, and Signal Processing, Proceedings. (ICASS). IEEE International Conference, 2002.
2. Hojoon Park, “A Method for Controlling the Mouse Movement using a Real Time Camera”, Brown University, Providence, RI, USA, Department of computer science, 2008.
3. Chu-Feng Lien, “Portable Vision-Based HCI – A Realtime Hand Mouse System on Handheld Devices”, National Taiwan University, Computer Science and Information Engineering Department.
4. Kamran Niyazi, Vikram Kumar, Swapnil Mahe, Swapnil Vyawahare, “Mouse Simulation Using Two Coloured Tapes”, Department of Computer Science, University of Pune, India, International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.2, March 2012.
5. K N. Shah, K R. Rathod and S. J. Agravat, “A survey on Human Computer Interaction Mechanism Using Finger Tracking” International Journal of Computer Trends and Technology, 7(3), 2014, 174-177.
6. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 2nd edition, Prentice Hall, Upper Saddle River, New Jersey, 07458.
7. Shahzad Malik, “Real-time Hand Tracking and Finger Tracking for Interaction”, CSC2503F Project Report, December 18, 2003.