iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** III **Month of publication:** March 2024

DOI: https://doi.org/10.22214/ijraset.2024.59268

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

Hand Motion: Mapping Gestures Through Recognition

G. Sanjay Gandhi[1], Shaik Nusrat[2], Ramavath Jhansi [3], Chandra Prakash Babu Ponduri[4], Samuel kumar Gollamudi[5]
[1]*Professor, Department of CSE, Vasireddy Venkatadri Institute of Technology (Autonomous), Guntur, Andhra Pradesh*
[2, 3, 4, 5]*UG Students, Department of CSE, Vasireddy Venkatadri Institute of Technology (Autonomous), Guntur, Andhra Pradesh*

Abstract: ***In our everyday dealings with friends, family and co-workers, gestures are crucial as they offer an easy and innate way of communicating. From simple hand movements to greet someone or acknowledge friends from a distance, gestures form part of our daily engagements. Everybody, recognizes gestures as the extension of our body language. If this concept is applied to human-computer interaction, it becomes possible to have hand gesture detection through vision-based technology. In this case, certain movements are performed before a camera with each movement corresponding to specific actions. For example, such motions can be recognized by a computer thereby rendering all these operations matching activities. Touchless interactive technology is developed through machine learning and computer vision advancements. With the challenges that come with the current pandemic in mind, there is a need for an artificial intelligence (AI) driven virtual computer mouse that can liberate us from the confines of a physical mouse and redefine traditional computer operation along practical and intuitive lines. This project commences by creating a gesture based virtual volume controller. We can set the stage for a time when hand recognition technology will be used exclusively for human-computer interaction by using a camera to recognize specified movements and then allocating those motions to actions related to typical mouse functions, such volume control. In order to meet the pandemic's concerns and further the development of interactive technology, it is imperative that we recognize and build upon the pioneering work of academics and researchers.***

Keywords: Computer visual perception, people and computers interaction, Recognizing hand motions, Technological interaction without direct physical contact, OpenCV, Hand tracking.

I. INTRODUCTION

The foremost edge of the technological advancements has been omitting physical human-computer interaction so as to reduce accessibility and make it easier. Examples include voice-controlled systems in vehicles, gesture-based living room game set-ups, including applications in video games, voice assistants and virtual reality simulations for comfort and entertainment among others. These advances are for comfort and entertainment but there are many cases when contactless interaction is needed or preferred over other methods. Let’s consider scenarios where factory workers or medical practitioners are busy with their hands engaged with completely different tasks. When such instances occur a system that has almost no touch can be a great advantage to professionals in the industry. With so much at stake, especially since we are still in the grip of the pandemic, minimizing transmission risks requires us to have contactless user interfaces between computers and humans. We aim thus at achieving this by developing a cheaply constructed system whose access is universal across all kind of uses his fictitious remedy turned out to be true, thanks to extraordinary progresses in computer vision and machine learning. Through these technologies we want to create a system that suits the current conditions of post pandemic era and as well addresses the wider need for effective human-machine interaction without physical touching across different sectors. continuous learning aimed at boosting hand detection confidence towards higher accuracy.

A. Hand Tracking Module

We started with designing a hand tracking module which should accurately recognize hand gestures and lay the foundation for more applications. Using Python’s computer vision and machine learning libraries, our goal was to achieve the best outcome. Another target has been to continuously learn so as to improve hand detection confidence for better accuracy.

B. OpenCV Contribution

OpenCV Library is used to recognize hand gestures for the purpose of controlling a PowerPoint presentation via an integrated webcam. In this code, the webcam is initialized while a hand detector is set up using cv zone library and gesture recognition parameters are defined.

©IJRASET: All Rights are Reserved

OpenCV's input in this work can be observed from its functions used for setting up webcam, image manipulation, and display. Using Hand Detector class from cv zone to detect hand landmarks and gestures illustrates what OpenCV can do in terms of computer vision. Furthermore, lines are drawn on the image using OpenCV functions that enable visualization of hand gestures as well as control of PowerPoint presentations accordingly. The role played by OpenCV's image processing capabilities in improving human-computer interaction with the slides cannot be underestimated.

C. Gesture-based Presentation Control

Gesture-Based Presentation Controller using hand gestures captured from the built-in webcam to control a PowerPoint presentation. It commences by initializing a PowerPoint presentation and bringing up the hand tracking camera. This controls features that track hands, captures frames upon frames, and then processes these frames to enable movement between slides. Specifically, a "thumbs up" gesture means go to the previous slide while a "whole hand pointing up" signal takes you to the next slide. Moreover, this code also includes a mechanism for drawing on slides using finger movements which can be set into motion through certain finger signals. The drawn notes are shown in the image frame. The main loop is responsible for continuously checking whether there were any gestures and updating the PowerPoint presentation accordingly. The q key terminates the program. Ultimately, this code represents an actualized demonstration of how computer vision and gesture recognition could be integrated for human-computer interaction during presentations.

D. Software Requirements

It depends on different libraries and specifications so as to allow hand gesture recognition for a slide presentation in human-computer interaction. The software requirements are as follows: enable control over slide navigation, the code uses win32com.client library for PowerPoint automation. It makes use of cv2 for webcam access and image processing, and CV Zone for hand detection and tracking within the captured images. Importation of external libraries like os and numpy is done to achieve general-purpose functionality. The camera set up parameters such as width, height, and gesture Threshold are used in defining the hand gesture recognition sensitivity. OpenCV (cv2) is used by the code to configure the webcam, specifically setting its width and height in terms of pixels. The Hand Detector class from cv zone is employed to detect and track a single hand from those images captured. Finger positions are analysed where "thumbs up" signifies backslide while "whole hand pointing up" represents forward slide. In addition, it shows an annotated current image frame whose termination is via pressing 'q' key on the keyboard by invoking this application. Make sure that you have installed required Python packages (cv2, cv zone, win32com) through proper package managers (pip).

II. LITERATURE SURVEY

In paper [1] the various challenges faced are variability of hand gestures, background clutter it needs to differentiate between the hand and the background objects. A technique of colour segmentation can be used and also depth sensing technique is also used.

The article written by Aashni Haria the system described in the research paper uses a camera to track hand movements. This system also can open websites, launch applications, and capable of controlling presentations. Despite the obstacles, this technique stands as a promising technology for HCI. With ongoing advancements, it's foreseeable that these kind of systems will evolve to offer higher accuracy, greater user-friendliness, and increased adoption. Moreover, beyond the stated challenges, it's worth acknowledging that the system outlined in the study paper has constraints in recognizing only a limited set of gestures. For hand gesture recognition to truly enhance HCI, future systems must expand their capability to identify a broader spectrum of gestures.

In the paper [5] it discusses various kinds of techniques, including: Shape analysis, appearance-based methods, motion analysis. The article also details some of the challenges associated with this kind of applications, such as: differences in hand appearance, disturbing backgrounds, Occlusions. The article concludes by discussing some of the future work directions, such as: developing more flexible and precise recognition algorithms, exploring new applications for communication between people and computer.

III. SYSTEM ARCHITECTURE AND DESIGN

A. Module Design

The voice control system has been split into several modules with different roles which we shall discuss as follows. One of which includes the voice-gesture recognition system that incorporates tracking and recognizing user's hand movements commonly referred to gestures such as pointing and thumbs ups to navigate through presentation slides. Whenever possible, this feature is most useful during presentations as one can move back or forward with their fingers without using a remote but through simple movements of hands in front of the camera.

Most importantly, this file must be called by every other file in order to use functions defined here when opening PPT images or running video annotation programs. The hand_detection.py file uses CV Zone hand tracking module for populating our cv2 window with all sorts of data like landmarks and positions of fingers used later on. Given these conditions, it becomes necessary for us to focus on how we will prepare our slideshows for errorless uploads. This is achieved by making sure that each slide contains information that can easily be understood by anyone who sees it regardless of their level or previous history in relation to this topic. In practice however, you may need your methods to have arguments. To do so, it uses annotations provided by python-pptx package. This tool provides two options: either an image saved into a modified .pptx document or video taken from webcam stream. In order to make sure that annotations always stay at the top even when slide changes generated dynamically are made throughout adding them was done automatically. In case there are any issues related to gesture detection troubleshooting might be helpful. It integrates well with the PowerPoint application and uses win32com.client to control your slides as well as to give a slide show that matches with the motions of your hands. The code readability, maintenance, and collaboration are all improved through this modular design. It makes easier the understanding of each module's specific responsibility, testing and altering individual components. When it comes to reusing code efficiently and scaling up, separation of concerns plays a crucial role.

IV. METHODOLOGY

A. Webcam Integration

As a built-in camera OpenCV is able to easily be integrated into an application. Through the use of the cv2.VideoCapture class, webcams can be initialized in order to ensure that frames are captured properly. Additionally, width and height parameters for the camera are set for the best possible capture resolution of images. Thereby, this specific arrangement makes it certain that the system operates with adequate clarity and responsiveness needed for correct hand gesture identification. In this regard, communications with webcam are established through OpenCV library. The object cv2.Capture Video Instantiation will serve as a foundation for capturing real-time frames. Since capture has to be customized according to particular specifications, such parameters as width and height can be adjusted. This careful setting up allows proper hand gesture recognition by the system by placing it at optimal settings for seamless user experience.

B. Hand Tracking Module

Application uses a simple webcam and gestures for controlling presentations. For example; during slide show navigation hand tracking is used which depends on CV Zone library. Lastly code employs win32com.client library to interact with Microsoft PowerPoint application. The built-in webcam is used by the application to capture video frames so that the user's hand movements can be detected and tracked using the "Hand Detector" class from CV Zone. The presentation has two different gestures for controlling: a thumbs-up gesture for going back to the previous slide and a whole hand pointing up gesture for showing the next slide. The win32com.client library loads up PowerPoint presentation, which is then started in slideshow mode. The camera is configured with a width and height of 900x720 pixels. In the main loop, video frames are continuously captured, and both hands and their landmarks are detected while specific gestures are being searched out. If there is any detected hand with known gesture, it will lead to corresponding action of either next or previous slide movement. A gestured threshold ensures that the hand is at an appropriate height for gesturing thus preventing accidental triggering. Moreover, instead of just drawing on the ongoing slide, the code additionally assists in annotations. In addition, once the user's hand is detected, the application allows for drawing lines on slide by tracing its movement across it. For example, the coordinates of the drawn lines are kept in an array list called annotations. Additionally, a counter and delay mechanism has been implemented to counteract multiple inputs that occur too fast or inadvertently. Once a gesture is recognized then there is a 30 frames delay (delay = 30) before another gesture can be detected. To sum up, this Python code combines computer vision with PowerPoint automation to develop a hands-free presentation control system that can also draw annotations onto slides through hand gestures.

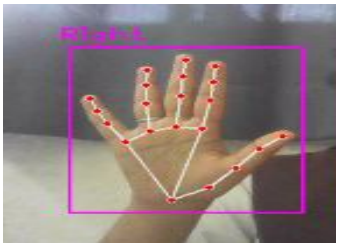


Fig. 1 Hand recognition

C. PowerPoint Integration

There is an implementation of a PowerPoint integration module. The main idea of this module is to allow control over presentations using webcams and hand movements, through the use of this technology hands free. Setting up the PowerPoint application involves two parts: Dispatching PowerPoint. Application from win32com.client and calling Presentations. Open() to open pptx, an existing presentation file. In effect, this allows Python script to communicate with the PowerPoint application. After setting up PowerPoint, video capture in real-time is enabled by the code's webcam configuration (cv2.VideoCapture). The dimensions of captured video frames are set at 900 x 720 pixels. A Hand Detector object which can detect hand landmarks and gestures in video stream based on CV Zone. Hand Tracking Module library is created. The heart of the system lies in interpreting certain hand gestures to control slides in PowerPoint. These code checks for gestures such as thumbs-up (fingers == [1,1,1,1,1]) for next slide and index finger raised (fingers == [1,0,0,0,0]) for previous slide. PowerPoint Presentation is controlled by the code using the methods of Next and Previous from the slideshows window of Presentation. Slide Show Window. View; this helps to move between slides in a smooth transition. One thing about the code is that it allows for annotation on slides. Consecutively annotated points are joined together with lines drawn on the current video frame OpenCV's cv2.line. This improves interaction during presentation because it makes it possible for the presenter to point out certain areas or further highlight vital information. System users control presentations through hand gestures which provide an easy-to-use and interactive way of navigating through slides. However, to prevent a rapid unintentional movement of slideshow, a delay function has been applied. Hand landmarks and annotations are overlaid onto real-time video coming from webcam cv2.imshow so that both speaker and audience will be able to view them. This application keeps running in an infinite loop, processing every video frame, detecting different gestures and changing PowerPoint presentation accordingly.

D. Gesture Recognition

The application has a fundamental aspect of gesture recognition that enables the user to navigate slides using specific hand movement. The system keeps track of the positions and features of the hands and it extracts meaningful information for identifying gestures. There are two types of gestures: Gesture 1 uses thumbs up to go back a slide; while Gesture 2 uses full hand pointing upwards, which advances the system to the next slide. This method ensures that power point presentation is controlled in an intuitive way. In implementation, the system utilizes finger landmarks and tracked hand position to recognize particular actions. Conditional statements check if the posture corresponds with Gesture 1 or Gesture 2. Consequently, once familiarized, commands are executed thus allowing seamless navigation in presentations. By doing this, we will be incorporating gesture recognition as one of its key components thereby enhancing its overall usability.

E. Slide Annotation

This feature of the application allows users to draw notes on the current slide directly. To achieve this, a list containing a set of points that represent the path drawn have to be created and manipulated. Annotations are formed by drawing lines which connect the points successively.

Users can thus animate their presentations with interactive annotations in real-time using it. The system keeps track of a list (annotations) which contains points that represent user's drawing movement. The cv2.line function is employed in connecting these points thereby making them appear as visuals for annotations. Through changing some functionality, one can control whether or not to draw annotation. This gives some interactivity to the presentation so that users may put annotations on screens dynamically while speaking.

F. User Interface

The user interface is a crucial part that displays real time images frames with annotated hand detections. OpenCV has been used as the library of choice to display the frames to provide an easy to visualize representation of interactions in the system. In this case, it ensures that web-cam frames are caught and handled continuously in order to enable chatting. In this implementation, OpenCV is used for presenting live image frames through cv2.imshow method. The structure of the loop contains capture and processing of each frame whereby making it possible for this application to be complete and dynamic. Interaction with users is made easier using the presented frames which allows slide navigation control without any breakages as well as annotation drawing. This implementation emphasizes on the necessity of an efficient and responsive user interface in giving an exciting/involving presentational experience.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

G. Optimizing Hand Detection

One of the major methods to optimizing hand detection is by fine-tuning detection parameters. It means that these parameters include confidence levels thresholds, non-maximum suppression and minimum detection size. Developers can adjust the sensitivity and specificity of hand detection algorithms to match application requirements by playing around with these parameters. Choosing Suitable Models and Libraries: Selecting proper hand detection models and libraries is vital for achieving best performance. Different computer vision frameworks have pre-trained models specifically tailored for hand detections tasks alone. Accuracy, speed, and resource efficiency are some of the things that should be considered when evaluating and comparing different models to identify the most suitable option for a given application. Data Augmentation and Training: In optimization, data augmentation and training techniques are essential in building custom hand detection models as well as developers. To increase model's robustness and generalization capabilities, it is important to augment training data with diverse hand poses, lighting conditions, as well as backgrounds. Also, iterative training steps can help improve on the accuracy of detections through fine-tuning model architectures as well as hyperparameters. Making Use of Hardware Acceleration: Among the benefits of using hardware acceleration technologies to perform hand detection are faster processing speeds. For instance, parallelizing on such computer accelerators as TPUs and GPUs can increase inference rates thus improving real-time performance for webcams. Including Post-Processing Methods: Moreover, post-processing can help improve the results of the hand detection system as well as avoid false positives and negatives. Some examples include but not limited to gesture filtering, hand tracking and temporal smoothing which enhances stability and accuracy of detected hand positions. Another improvement is adding spatial constraints based on anatomy of a human hand to make it more reliable.

V. OUTPUT

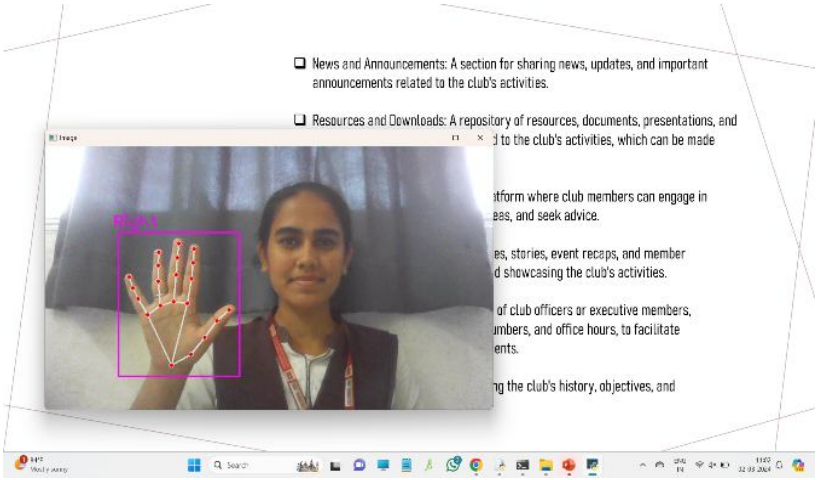


Fig. 2 Hand gesture for moving to next slide (right hand)

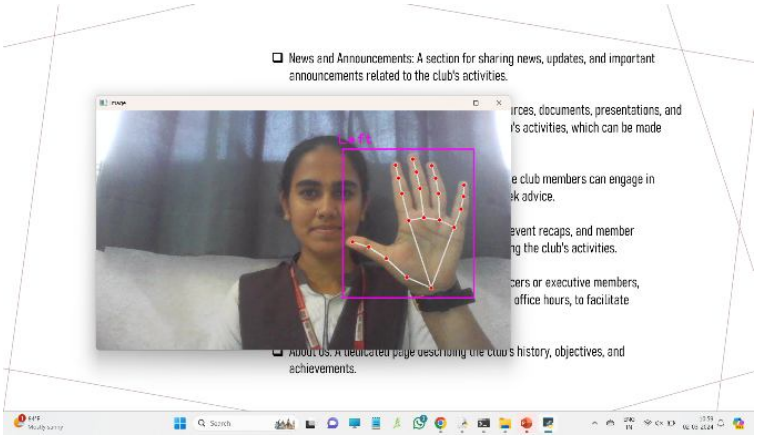


Fig. 3 Hand gesture for moving to next slide (left hand)

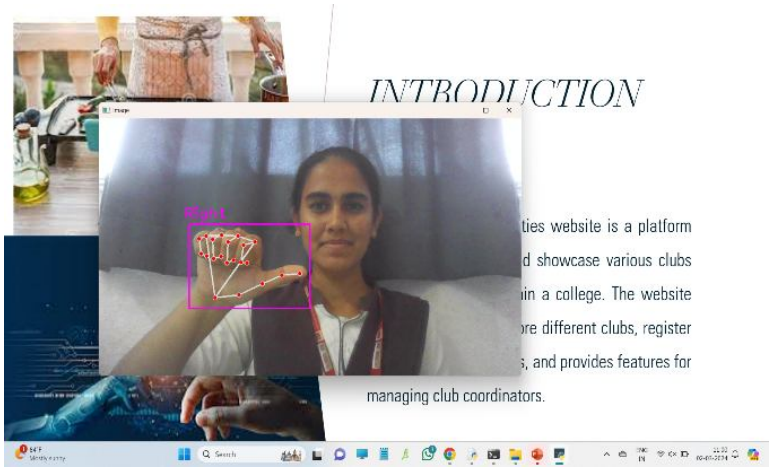


Fig. 4 Hand gesture for moving to previous slide (right hand)

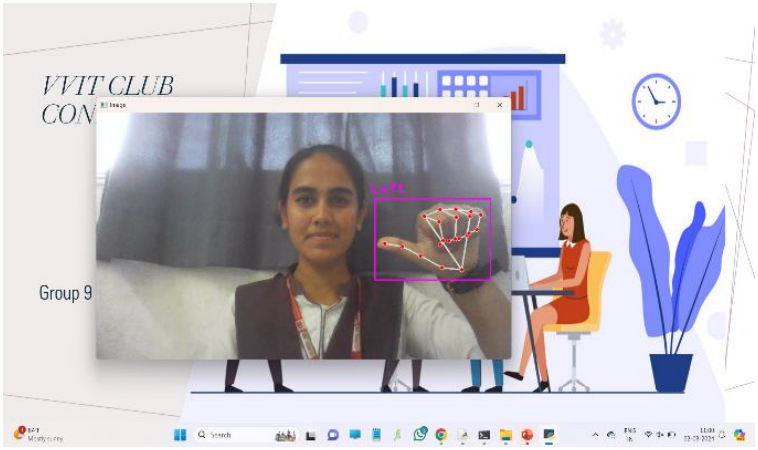


Fig. 5 Hand gesture for moving to previous slide (left hand)

VI. RESULTS AND DISCUSSIONS

This module detects the user's hand and its landmarks which allows for recognition of specific gestures. It functions by using a webcam, while the slides are loaded from a predefined PowerPoint file. Gesture based slide navigation is a feature that makes this application unique among others. The system moves on to the next slide once Gesture 2 is recognized (whole hand pointing upwards). Conversely, when Gesture 1 (thumbs up) is detected, the system navigates to the previous slide. This algorithm works only when the hand is at face height so as to avoid accidental inputs. Another feature is slide annotation. The user can enter a drawing mode by making this particular gesture and then draw on the slides with his/her hands. Annotations made remain on screen until the next slide transition thus giving an interactive presentation experience. The most important thing about this app is that it prioritizes interaction and quick response time. To ensure smooth control for presenting in class or other occasion without any interference caused by many gestures being registered within a small time gap, there is a delay mechanism in place to cater for multiple gestures in short period of time. The system restarts after some specified time allowing future recognized direction signals.

VII. CHALLENGES

It is dependent on third-party libraries, such as Aspose, CV Zone, and Win32Com. This can be challenging, especially if the user has no idea about these libraries. This assumes specific library versions and assumes that the operating system (Windows) is compatible. Compatibility with different library versions and various types of operational systems may be hard to establish. This hard-codes other file locations as well as paths for the PowerPoint presentation. If this code runs on another machine with different file directories, this could pose a problem. The program does not contain comprehensive error management. For real-life cases, unexpected errors such as missing files or wrong settings may crash it. Any application needs robust error handling techniques to become reliable.

Recognition by the gesture recognition is limited to thumbs up and pointing up only. More gestures can be supported for better user experience with the system or its resistance to alterations in gestures improve to become more stronger. Application's real-time performance is determined by the effectiveness of the hand tracking module and how fast your computer processes the data. It might be hard to insure prompt and smooth actions especially in various lighting situations. This program uses a webcam without authorization or authentication. There could be security issues here, so it's necessary to ensure user verification to prevent unauthorized access. There is a possibility that the code will not effectively release resources like PowerPoint objects and camera capture. Proper resource management must be done in order to avoid memory leaks and provide stability for long period use of the application.

VIII. FUTURE ENHANCEMENT

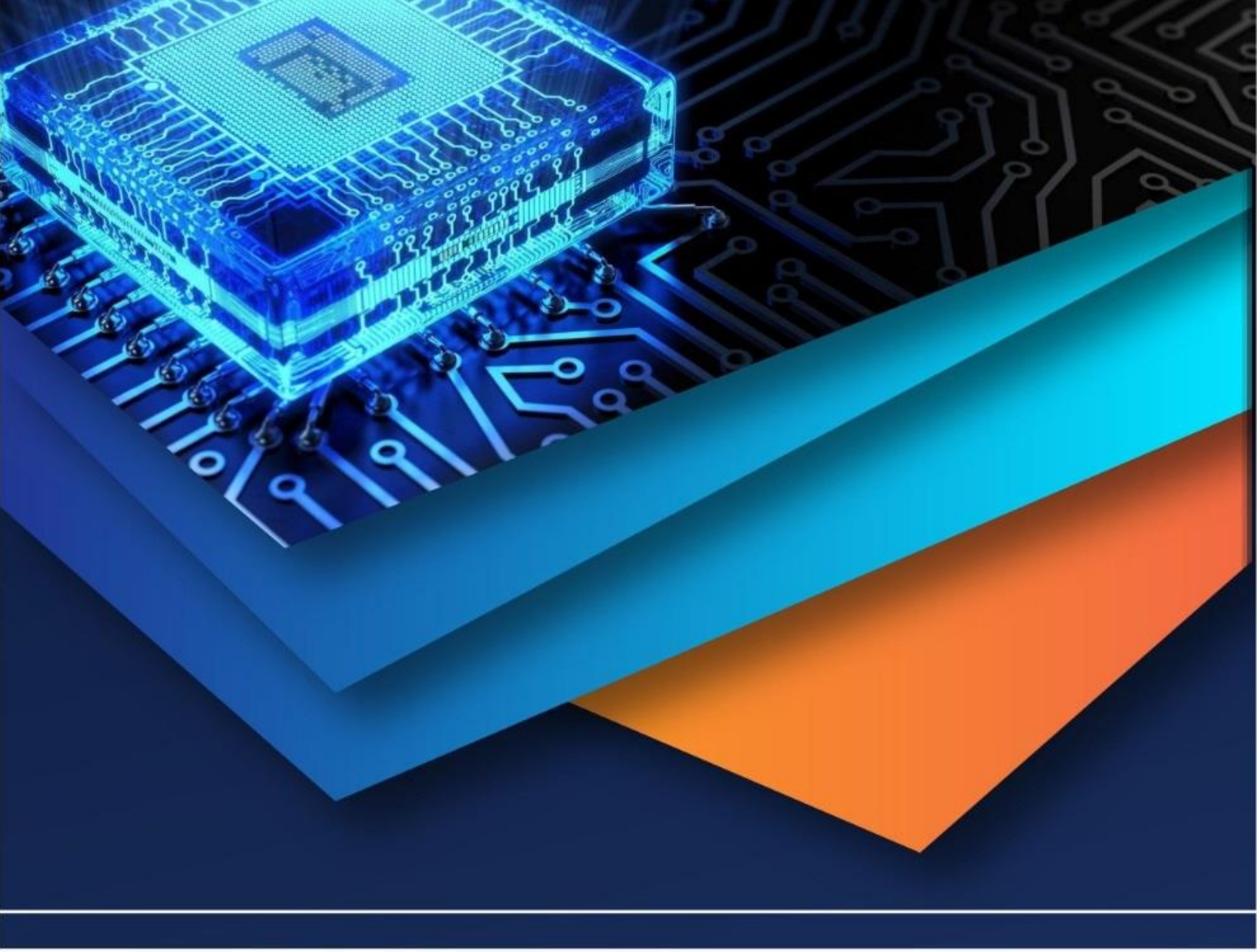
Expand gesture recognition in order to include a wide range of hand movements for multiple tasks. For instance, introduce gestures that can open/close presentations, blank the screen or switch between programs. It should be possible to configure the system such that users are capable of defining their own gestures for particular activities. Integrate voice command recognition so that there is another input option or as an enhancement. This permits users to employ their voices while performing actions on slides and other functions instead of just relying on touchless gestures only. Additionally, add a feature that displays slide thumbnails thus making it easier for the presenter to see and move through slides quickly. Moreover, integrate gestures for choosing specific slides from these thumbnail views. Implement calibration sequences allowing customization and fine tuning by users according to hand sizes and corresponding gestures. Provision of plug-and-play configuration ensures accurate personalized gesture recognition. Also extend the application to accommodate simultaneous use by many people. This allows collaborative presentations where each user can independently manipulate a presentation via their own motions.

It can improve the user interface by giving informative responses on gestural recognition, connection status and possible errors. Add error handling code that gracefully handles unforeseen circumstances. Enable cloud services integration for data storage and retrieval of presentation data. This will make it easier for presentations to be accessed across multiple devices by allowing seamless collaboration. Develop a mobile app companion that connects to the presentation system. This mobile app can be used as a remote control, whereby people can control their presentations from either smartphones or tablets.

REFERENCES

[1] Sundus Munir, Shiza Mustaq Afrozah Nadeem, Syedabinish Zahra, "Hand Gesture Recognition: A Review", International Journal of Scientific & Technology Research 10(05):392, July 2022.
[2] Hajeera Khanum and Dr. Pramod H B, "Smart Presentation Control by Hand Gestures Using Computer Vision and Google's Mediapipe", International Research Journal of Engineering and Technology (IRJET), volume 09,issue 07, July 2022.
[3] Kavin Chandar Arthanari Eswaram, Akshat Prakash Srivastava, and M. Gayathri "Hand Gesture recognition for Human-Computer Interaction Using Computer Vision" March 2023.
[4] Bhor Rutika, Chaskar Shweta, Date Shraddha, Prof. Auti M. A. "Power Point Presentation Control Using Hand Gestures Recognition" , International Journal of Research Publication and Reviews, Vol 4, no 5. May 2023.
[5] Prof. Nilam Honmane1 , Ashitosh Nevase2 , Sourabh Pawar3 , Vaibhav Purane4 , Saurabh Sawant5 "Hand Gesture Recognition System Using Deep Learning", International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), Volume 3, Issue 16, May 2023.
[6] Munir Oudah, Ali Al-Naji and Javaan Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques", J Imaging, 2020 Aug.
[7] Jerald siby, Hilwa Kader and Jinsha Jose, "Hand Gesture Recognition", (IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH Volume No.3, Issue No.2, February – March 2015.
[8] Prof. Nilam Honmane, Ashitosh Nevase, Sourabh Pawar, Vaibhav Purane, Saurabh Sawant, "Hand Gesture Recognition System Using Deep Learning", International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), Volume 3, Issue 16, May 2023.
[9] Soukaina Chraa Mesbahi, Mohamed Adnane Mahraz, Jamal Riffi, Hamid Tairi, "Hand Gesture Recognition Based on Various Deep Learning YOLO Models", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 14, No. 4, 2023.
[10] Ahmad Puad Ismail, Farah Athirah Abd Aziz, Nazirah Mohamat Kasim and Kamarulazhar Daud, "Hand gesture recognition on python and opencv", IOP Conf. Series: Materials Science and Engineering 1045 (2021) 012043.
[11] Ruchi Manish Gurav and Premanand K. Kadbe, "Real time Finger Tracking and Contour Detection for Gesture Recognition using OpenCV", International Conference on Industrial Instrumentation and Control (ICIC) College of Engineering Pune, India. May 28-30, 2015.
[12] Rhio Sutoyo , Bellinda Prayoga, Fifilia, Dewi Suryani, Muhsin Shodiq, "The Implementation of Hand Detection and Recognition to Help Presentation Processes", International Conference on Computer Science and Computational Intelligence (ICCSCI 2015).
[13] Arpit Mittal, Andrew Zisserman, Philip H. S. Torr, "Hand detection using multiple proposals", MITTAL et al.: HAND DETECTION USING MULTIPLE PROPOSALS.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)