Computer Cursor Control using Facial Gestures for Physically Challenged

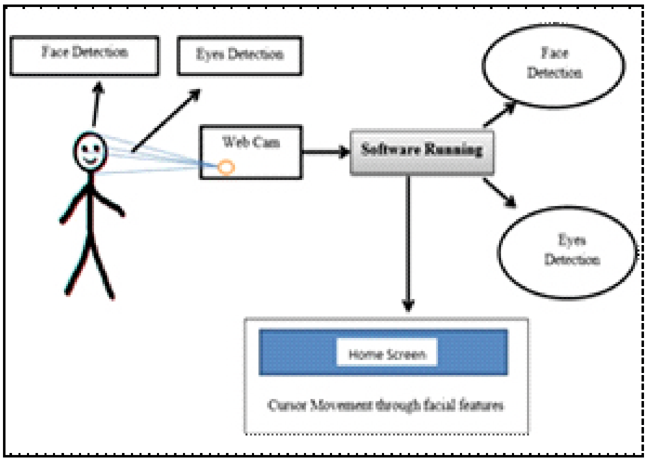
Dr.Saroja S Bhusare , Dr.Veeramma Yathnalli, Prerana D Swamy, Pruthvi K, Rakshita M, S Meenakshi
Dept of ECE line 3: JSS Academy of Technical Education, Bangalore, India
Email : sarojasbhusare@jssateb.ac.in, veerammayatnalli@jssateb.ac.in, preranadinakar29@gmail.com
pruthvik242@gmail.com, rakshitamahendrakar1234@gmail.com, meenakshis58765@gmail.com

***Abstract* - Nowadays everything are in digital transformation. And the need for this has grown to a large extent. Human computer interaction (HCI) system is of great importance to amputees and those who have issues with using their hands. Our project provides a hand-free interface between human and computer which is of great use to amputees. Here basically we have replaced the use of mouse by the movements of eyes. Here in our project various image processing methods such as face detection, eye extraction and interpretation of a sequence of eye blinks in real time for controlling human computer interface has been used. We have used a webcam to capture an input image. When the webcam captures the image according to the movement of eyes like to right, left, up and down the cursor moves. For enabling other actions for cursor, blink option is used to activate select option. To perform these operations CNN, Dlib, Haar cascade algorithm is being used . This system is mainly aimed for physically challenged to have an effective communication with computer.**

1. INTRODUCTION

HCI-Human Computer Interface. The HCI system is basically focused to provide interaction between computer and human. And presently we have a need to find new technology that creates effective communication between human and computer. Here human computer interaction plays an important role to this day. Thus there is need for facilitating communication between humans and computers to promote further advancement. Nowadays as computer- aided learning is growing up, the significance of human computer interaction is rapidly expanding. There is need for Personal and computer computations in the workspace as well as for academic purposes. This will be very much important for paralyzed person, physically challenged people and especially people without hands. Here a webcam is used to capture the image first. Before proceeding further the image will be preprocessed. The preprocess procedure includes flipping the image and converting it into grayscale image. Further these movements are drawn to computer by

using the cartesian method. By using this method ,landmark points are marked and further actions are carried out. Mouse movement is adjusted according to the code written and defined ratios. According to the ratios defined in the code movements of mouse is controlled. Here the system will take the real-time video input from the user with the help of OpenCV and run it in the background. And after capturing the image from webcam, by the movements of eyes the cursor is controlled.



2. IMPLEMENTATION

Execution of the whole code is done via PyCharm. In addition to this many packages are installed to perform various operations.

PyCharm

PyCharm is an Integrated Development Environment (IDE) specifically designed for Python development. It is developed by JetBrains and provides a powerful and feature- rich environment for writing, debugging, and running Python code. PyCharm offers a range of tools and features that enhance productivity and facilitate the development process for Python programmers.

OpenCV

The "cv2" package in Python refers to the OpenCV library, which stands for Open Source Computer Vision Library. OpenCV is a popular open-source computer vision and image processing library that provides a wide range of functions and algorithms for tasks such as image and video processing, object detection and tracking, and more.

To use the "cv2" package in Python, you'll need to install OpenCV first. You can install it using pip, the package installer for Python, by running the following command: Pip install opencv-python

Once installed, you can import the "cv2" module in your Python script and use its functions and classes to perform various computer vision tasks.

Mediapipe

The "mediapipe" package in Python refers to the MediaPipe framework, which is an open-source framework developed by Google for building pipelines for complex multimodal machine learning (ML) workflows. MediaPipe provides a collection of pre- built components and tools for processing and analyzing multimedia data, including video, audio, and 3D data. MediaPipe offers various pre-built solutions, known as "graphs," that you can use for tasks such as hand tracking, pose estimation, face detection, and more.

Time

The "time" package in Python is a built-in module that provides various functions related to time. It allows you to measure time, work with timestamps, and control the execution speed of your code.

Math

The "math" package in Python is a built-in module that provides various mathematical functions and constants. It allows you to perform mathematical operations, manipulate numbers, and access mathematical constants.

NumPy

The "numpy" package in Python is a powerful library for numerical computing. It provides a multidimensional array object, various mathematical functions, and tools for working with arrays. Numpy is widely used in scientific computing, data analysis, and machine learning tasks.

PyautoGui

The "pyautogui" package in Python is a cross-platform GUI automation library that allows you to control the mouse and keyboard to automate tasks on your computer. It provides functions for simulating mouse clicks, movements, and keyboard inputs, as well as functions for taking screenshots and interacting with the screen.

PyWhatkit

Python offers numerous inbuilt libraries to ease our work. Among them pywhatkit is a Python library for sending WhatsApp messages at a certain time, it has several other features too.

3. METHODOLOGY

The methodology is as follows:

1) Since the project is based on detecting the features of the face and mapping them to the cursor, the webcam needs to be accessed first, which means that the webcam will be opened. Once the webcam is opened, the program needs to extract every frame from the video. The frame-rate of the video is generally around 30 frames per second, so a frame at every 1/30th of a second will be used to be processed. This frame undergoes a set of processes before the features of the frame are detected and mapped to the cursor. And this process continuously takes place for every frame as a part of a loop.

2) Once the frame is extracted, the regions of the face need to be detected. Hence, the frames will undergo a set of image-processing functions to process the frame in a suitable way, so that it is easy for the program to detect the features such as eyes, mouth, nose, etc. The processing techniques are:

✦ Resizing: The image is first flipped over the y-axis. Next, the image needs to be resized. The resize function refers to setting the new resolution of the image to any value as per the requirement. In this project, the new resolution is 640X 480.

✦ BGR to Gray: The data that we are using to detect the different parts of the face requires image of a grayscale format to give more accurate results. Hence, the image, i.e the frame of the video from the webcam needs to undergo the process of converting its format from RGB to grayscale. Once the image is converted to a grayscale format, it can be used to locate the face and identify the features of the face.

✦ Detection and Prediction of facial features: To detect the face and the features, a prebuilt model is used in the project, which has the available values that can be interpreted by python to make sure that the face is located in the image. There is a function called 'detector()', made available by the models, which helps us to detect the face. After the face is detected, the features of the face can now be detected using the function 'predictor'. The function helps us to locate 68 points on any 2D image. These points correspond to different points on the face near the required parts such as eyes, mouth, etc. The values of the function that are obtained are in the form of 2D coordinates. Every one of the 68 points are essentially values of the x and y coordinates that, when connected, will roughly form an identifiable face. They are then stored as an array of values so that they can be arranged and used in the next step to connect any of the coordinates and draw a boundary to represent the required regions of the face. Four sets of arrays are taken as 4 different parts of these values which are stored in the array, to separately be stored as the coordinates to be connected to represent the required regions, those are the: Left eye, Right eye and hand gestures. Once these arrays are prepared, boundaries, or 'contours' are drawn around the points using 3 of these arrays by connecting these points, using the 'draw contour' function and the shape formed is around the two eyes and the hand.

- Eye aspect ratios: Once the contours are drawn, it is necessary to have a reference for the shapes, which, when compared with, gives the program any information about any action made by these regions such as blinking, moving, etc. These references are understood as ratios, between the 2D coordinates, and a change in the coordinates, essentially tell us that, the parts of the region of the face have movedfrom the regular position and an action has been performed. The system is built on predicting facial landmarks of the face. The Dlib prebuilt model helps in fast and accurate face detection along with 68 2D facial landmarks as explained already. Here, Eye-Aspect-Ratio (EAR) are used to detect blinking/winking and moving respectively. These actions are translated into mouse actions.

The graph shows that EAR value drops drastically when the eye is closed. This trigger can be used for clicking action. Hence, for these functionalities to be made operational, there need to be some defined 'aspect-ratios', which when cross a defined limit, interprets an action being performed

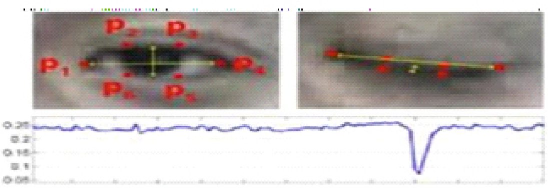


Fig – 4.4 : Eye aspect ratio

$$\text{EAR} = \frac{||P2 - P6|| + ||P3 - P5||}{||P1 - P4||}$$

- Detection of actions performed by the face: After the ratios are defined, the frame can now compare the ratios of the parts of the face with the ratios defined for different actions, of the current frame being processed. It is done using the 'if' statement. The actions which the program identifies are:

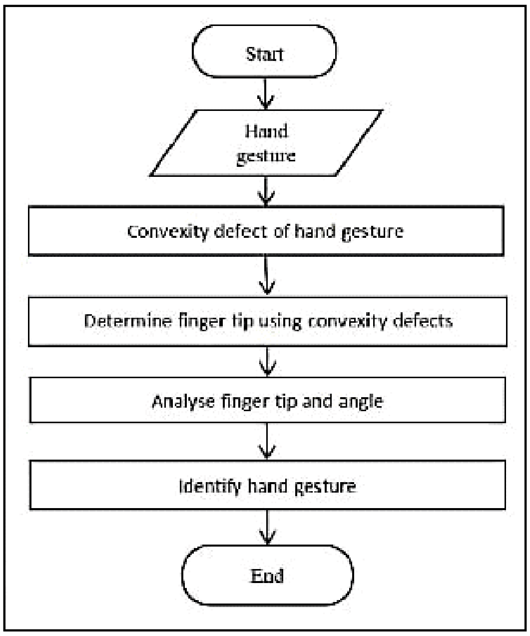
1) Left/Right Clicking: For clicking, he needs to close any one of his eye, and make sure to keep the other open. The program first checks whether the magnitude of the difference is greater than the prescribed threshold by using the difference between the ratios of the two eyes, to make sure that the user wants to perform either the left or right click, and does not want to scroll (For which both the eyes need to squint).

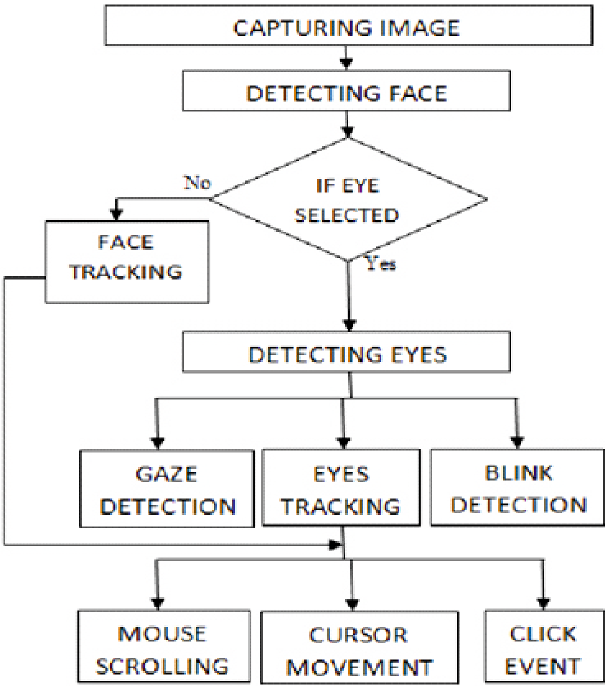
2) Scrolling: The user can scroll the mouse, either upwards or downwards. He needs to squint his eyes in such a way that the aspect ratio of both the eyes is less than the prescribed value. In this case, when the user places his nose outside the rectangle, the mouse performs scroll function, rather than moving the cursor. He can move his nose either above the rectangle to scroll upwards, or move it below the rectangle to scroll downwards.

3) Volume control: Detect hand landmarks. Calculate the distance between thumb tip and index finger tip. Map the distance of thumb tip and index finger tip with volume range. For my case, distance between thumb tip and index finger tip was within the range of 30 – 350 and the volume range was from -63.5 – 0.0. In order to exit press 'Spacebar'

The below figure shows the working of hand gestures. The system is trained using a dataset of hand gestures to recognize different gestures. Once the gesture is recognized, it is translated into a corresponding mouse movement, which is then executed on the virtual screen. The system is designed to be scalable and adaptable to different types of environments and devices.

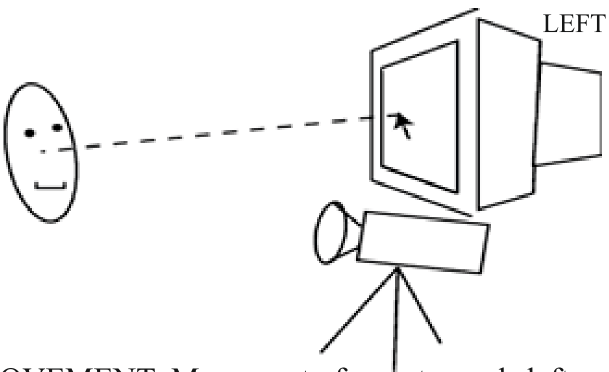
Another figure depicts the working of facial gesture controlled cursor movement. The mouse cursor can be moved by making facial movements to the left, right, up, and down, and mouse events are controlled by blinking the eyes.



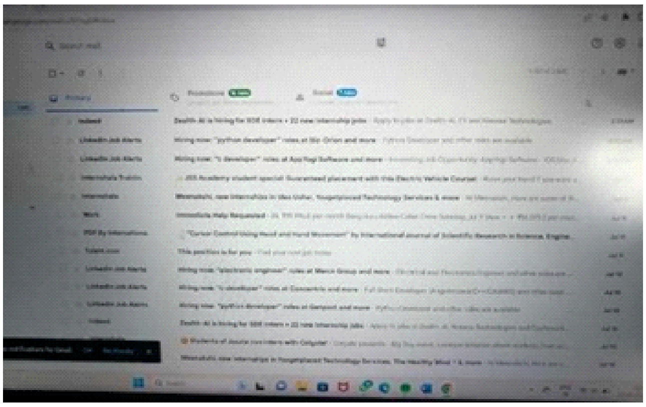


4. RESULTS

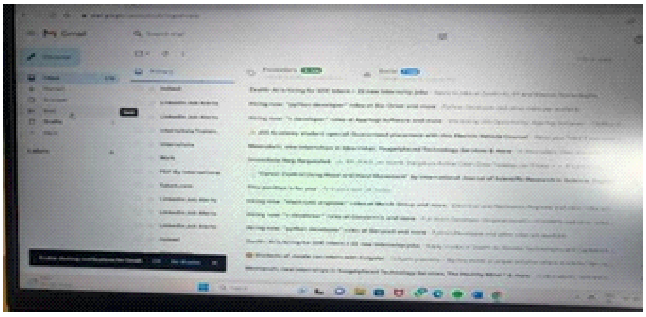
FACE: As the code gets executed the camera turns on and cartesian are marked on the face. Various parts like eyes, mouth, nose, ears are detected and mapped.



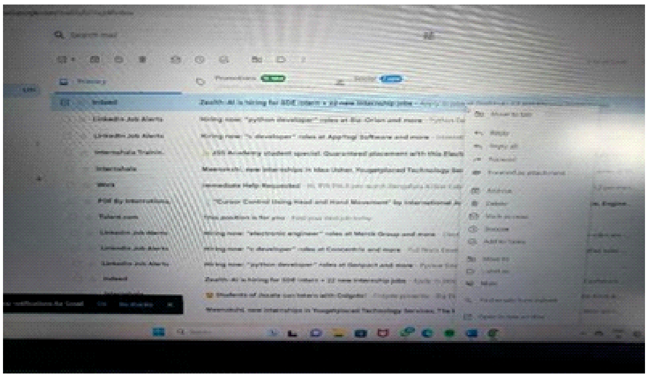
MOVEMENT: Movement of eyes towards left initiates right movement in the cursor



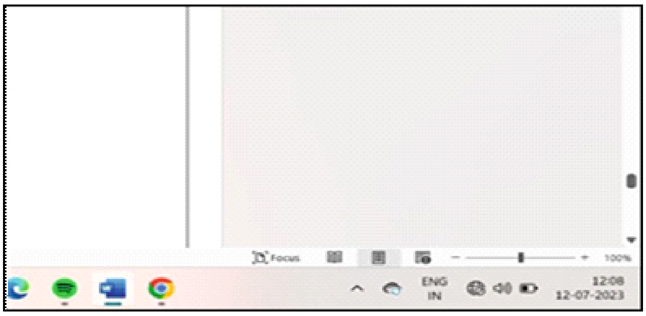
RIGHT MOVEMENT: Movement of eyes towards right initiates left movement in the cursor



BLINK: Upon blinking left eye or both eyes for a particular period of time select option is enabled



VOLUME CONTROL: If the distance between the index finger and the thumb finger is increased, the volume gets high. If the distance between the index finger and the thumb finger is decreased, the volume gets low.



5. CONCLUSION

Our system is mainly focused on physically challenged and paralyzed people to compute efficiently and easily. An interaction between computer and mouse is replaced with human eye movement. It uses typical web cam to capture an input image. In conclusion, computer cursor control using facial gestures represents an exciting and evolving field within human-computer interaction. The ability to navigate and interact with digital interfaces through natural facial expressions and movements has the potential to enhance user experiences, accessibility, and convenience. As of my last knowledge update in September 2021, significant progress had been made in this area, with various research papers and studies showcasing innovative approaches and applications.

Gesture Recognition Advancements: Researchers have developed increasingly sophisticated techniques for recognizing and interpreting facial gestures in real-time. These methods often employ computer vision, machine learning, and deep learning algorithms to accurately detect and classify gestures.

User Accessibility: Facial gesture-based cursor control has the potential to make computing more accessible to individuals with disabilities, such as those with limited mobility or dexterity. It can provide an alternative means of interaction that is more intuitive and inclusive.

Natural User Interfaces: By using facial gestures, users can interact with computers in a more natural and intuitive way, reducing the reliance on traditional input devices like mice and keyboards. This can lead to more immersive and engaging user experiences.

Application Areas: Facial gesture-based cursor control has been explored in various applications, including gaming, virtual reality, healthcare, and education. It offers potentialbenefits in scenarios where hands-free or touchless interaction is desirable.

Privacy and Ethical Considerations: As with any technology involving facial recognition, privacy and ethical concerns must be addressed. Safeguarding user data and ensuring consent and transparency in data collection and usage are critical considerations.

Future Directions: The field continues to evolve, and future research may lead to more accurate and robust gesture recognition systems. Advances in hardware and software are likely to drive further innovation in this area.

In summary, computer cursor control using facial gestures holds great promise for improving human-computer interaction and accessibility. It represents a shift toward more natural and intuitive ways of engaging with digital technology.

Our project mainly focuses on integrating both facial and hand gestures. So by implementing the software the entire project runs. Our system was basically focused on how both the implementation can be collaborated. Here both the facial and hand gesture works simultaneously. By running the program whichever option enabled ,it works accordingly.

User-Centric Design: Hand gesture-based cursor control prioritizes user experience and convenience. It can enhance the accessibility of digital devices for individuals with mobility impairments, offering an alternative and inclusive interaction method.

Versatile Applications: Hand gesture control has found applications in diverse fields, including gaming, virtual reality, augmented reality, healthcare, automotive interfaces, and smart home systems. Its versatility allows for creative and innovative user experiences.

Privacy and Security: Similar to facial gesture recognition, hand gesture recognition systems must address privacy and security concerns. Proper data handling, user consent, and protection against unauthorized access are essential considerations.

Integration with Existing Interfaces: Integrating hand gesture control seamlessly with existing computer interfaces remains an ongoing challenge. Ensuring a smooth transition and coexistence with traditional input methods is crucial.

REFERENCES

[1] Shashidhar R; Snehith K; Abhishek P K; Abhishek B Vishwagna , Mouse cursor Control Using Facial Movements - An HCI Application" , IEEE- 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 27 April 2022

[2] Vinay S Vasisht, Swaroop Joshi, Shashidhar, Shreedhar , " Human computer interaction

based eye controlled mouse" , IEEE- 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 02 September 2019.

[3] Vinay S Vasisht ,Swaroop Joshi, Shashidhar, Shreedhar, "Human computer interaction based eye controlled mouse" , IEEE- Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology [ICECA 2019].

[4] Tereza Soukupova and Jan Cˇ ech. "Real-Time Eye Blink Detection using Facial Landmarks" In 21stComputer Vision Winter Workshop, February 2016.

[5] Hari Singh and Jaswinder Singh, "Real-time eye blink and wink detection for object selection in HCI" in part of Springer Nature, + Springer International Publishing AG, 2018.

[6] "Facial Gesture Recognition for Cursor Control" By Richard D. Moss, David G. Barnes, and Mohammad R. Neshatian .Published in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2010.

[7] "Real-Time Face Gesture Recognition for Human-Computer Interaction" by Shiqi Li, Zhenqiang Ying, and Weihua Ou.Published in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2011.

[8] "FaceMouse: A Human-Computer Interface for Mouse Control Using Real-Time Face Tracking" by Weihua Ou, Shiqi Li, and Zhenqiang Ying.Published in: IEEE Transactions on Cybernetics, 2012.

[9] "Deep Learning for Real-Time Facial Gesture Recognition Using a Webcam" by Muhammad Usama and Muhammad Usama Maqsood Published in: 2019 International Conference on Robotics and Automation (ICRA), 2019.

[10] "Real-Time Facial Gesture Recognition for Human-Computer Interaction" by Yan Sun, Liang Xie, and Tianwei Zhang Published in: IEEE Transactions on Cybernetics, 2015.

***