Computer Cursor Control Using Eye and Face Gestures

Akshada Dongre
Department of Information Technology
St. Francis Institute of Technology
Mumbai, India
dongre.akshada@gmail.com

Rodney Pinto
Department of Information Technology
St. Francis Institute of Technology
Mumbai, India
pintorodney198@gmail.com

Ameya Patkar
Department of Information Technology
St. Francis Institute of Technology
Mumbai, India
ameyapatkar08@gmail.com

Dr. Minal Lopes
Department of Information Technology
St. Francis Institute of Technology
Mumbai, India
minallopes@sfit.ac.in

Abstract — **In this paper, a computer application is developed using Python and OpenCV. This application will make it possible for the user to move the cursor point on the screen of the computer and also additional functionalities like clicking on the computer screen. The proposed system is a virtual interactive module for the users and can be used as an alternative module of touch screen technology. In this system, after the input is taken from the user, the real-time video input is processed and face recognition is performed. A point on the user's face would control the cursor on the screen and the right and left wink would implement the right and left click respectively. Squeezed eyes would facilitate the enabling of the scroll function in case of reading PDFs and other documents. It would thus facilitate the use and movement of the cursor through the face gestures. The application can be implemented on desktops and laptops. With the help of this application, the user can perform actions like move the cursor in all directions, scroll, drag and click. Thus the system would be extremely helpful for physically challenged users.**

Keywords— cursor movement, aspect ratio, click operation, cursor control, HCI

I. INTRODUCTION

As computer aided learning is growing up, the significance of human-computer interaction is rapidly expanding. Human and computer interconnection has expanded in recent years. Personal and computer computations are a need in the workspace as well as for academic purposes. Thus, a vision-based approach is taken into account and an effective technique to develop human-computer interface systems is used. A webcam is required to acquire images of facial movements that are recorded by the webcam. These movements are further graphed to a computer screen to position a mouse cursor accordingly.

The movement of the mouse is automatically adjusted by the position of the anchor point. Camera is used to capture the image of face movement. The system would take the real-time video input from the user with the help of OpenCV and run in the background. The input images taken from the user are converted from RGB images to grayscale images; a certain point on the user's face would control the cursor on the screen and the right and left wink would implement the right and left click respectively. Squeezed eyes would facilitate the enabling of the scroll function in case of reading PDFs and other documents. It would thus facilitate the use and movement of the cursor through the face gestures.

II. STATE OF ART

Many researchers have contributed the work supporting eye tracking in several domains. They have used different systems and technology to trace the eyeball of the user. In reference [1], the author tried to work on the system's slow response to eye detection. A processor namely, Pentium 11333 Mhz, machine is used for the detection of pupil and the face. This includes taking 30 frames per second. Moreover, it makes use of a PCI frame grabber that is incapable of processing the acquired image and therefore synchronization of the frames is done with the help of a hardware that illuminates the pupil. In [2] the paper addresses a real-time detection of the face during which each scale and location within each image is analyzed for real-time speed. Color filtering or motion filtering in video clips are also used by other devices. Color and movement in broad databases are not readily accessible.

Vision based eye tracking is the clear picture of the attention tracking movement described on how the movements of the eyes were done and the way the tracking of the eyes together with the gaze movements accustomed occur. The instance of eye gaze provides context to the pc system that the user is watching and so supports effective interaction with the user systems and is accustomed to measure visual attention and to facilitate different sorts of user interaction tasks [3]. The system utilizes robust eye-tracking data from multiple cameras to estimate 3D head orientation via triangulation. Multiple cameras also afford a bigger tracking volume than is feasible with a private sensor. The obtained results demonstrate the effectiveness of the top pose calculation subsystem, both applications use head pose data within the type of regions gazed by the user, verification is done based on whether gaze estimates by a people are the identical as those actually estimated by the system. The results are given within the type of a percentage of correct recognition by our tracking system.

In [4], The author has used OpenCV for Real-time video processing within the android smartphone. Here the author is trying to create two Android applications supporting video processing using two different methods; one by using OpenCV library, the second one is using an Android library with a self-implemented algorithm called ‘CamTest’. Here the author has used eight image recognition techniques to test the efficiency of the Android library and OpenCV in every frame of video captured on an Android smartphone. With above techniques, the paper concludes that most image processing techniques using the OpenCV library perform higher than the self-made Android library algorithm, and that OpenCV gives greater focus to improvement than energy consumption. Different approaches are used for eye detection. In [5], the author has explained the three different approaches. The Regression approach is used to minimize the distance between the predicted and actual eye positions,Bayesian approach learns model of eye appearance and non-eye appearance. and Discriminative approach Treats the problem as one of classification.

In [6] and [7], computer graphics and OpenCV are used for Fatigue Detection using smartphones and facial behaviour detection using machine learning approaches, the author used Computer Vision for the appliance. This study contributes to the problem of coming across driver fatigue through supplying a operating prototype for real-time fatigue detection and the usage of an Android smartphone or an iPhone. The smartphone-based fatigue detection science in lowering drowsiness-related traffic accidents and enhancing riding safety. The National Sleep Establishment evaluated in 2002 that 51 percent of grown-up drivers had driven a vehicle while tired and 17 percent had fallen asleep at the wheel. To scale back the dangers of driver weariness, a smartphone-based innovation is created to watch visual pointers of driver weakness, counting head gestures, head revolutions, and eye squints. This smartphone-based weariness location innovation gives a transportable and reasonable elective to existing weariness location frameworks. This article uses smartphones to observe visual driver fatigue indicators and facial behaviours which enable the possibility of building fatigue detection systems. This fatigue detection technology on smartphones provides a viable and affordable alternative to existing fatigue detection systems. Computer vision innovation involves a comprehensive application for driving safety improvement. Many of the safety techniques (for instance drowsiness detection, forward stability control and automatic braking alarms) are reliant on Machine Learning Algorithms. Drowsiness detection is done with the help of ML techniques like Mouth Opening Ratio and Nose Length Ratio. A virtual drive experiment to investigate the visual signs of driver fatigue was circulated in order to identify the efficacy of smartphone-based driver fatigue detection. Prior to the actual driving mission, participants were screened.For order to be eligible as candidates, candidates will have a minimal experience of two years, regular or modified vision, no target-indications, such as disease, intoxication or treatment. Caffeine and tea for 4 hours and alcohol for 24 hours until the analysis were not questioned. The iPhone and visualization software save head nodes, eye blinks and head rotations that are included in the visual behaviour.

It is observed that the technologies used in reviewed literature require excessive lighting conditions. Also, the performance measure drops in RGB Image format. The existing systems have used various technologies for fatigue detection and eye detection and require high pixel rate for cameras to track the face of users. This paper proposes a system for detecting and monitoring facial gestures using the dlib library. The various motions on the screen are carried with the help of input taken from the user and the real-time video input is processed and face recognition is performed. The input images are transformed into grayscale images and are further used to track the movements on the users face. It would thus facilitate the use and movement of the cursor through the face gestures.

III. PROBLEM IDENTIFICATION

The terrific venture lies in growing an economically feasible and hardware-independent machine in order that humans can engage except having any bodily reference to the PC. The bold block lies in creating a system that is environment friendly to use and at the identical time, it is effortless for a couple of types of users. It needs to supply flexibility to the person as properly as it needs to no longer be time-consuming. The most goal is to enhance an object monitoring utility to interact with the pc, and a digital human-computer interaction device, whereby no bodily contact with the machine is required and additionally to produce a higher human-machine interplay routine. To develop a user-friendly system for physically challenged people who might be accustomed to perform tasks easily.

A. Abbreviations and Acronyms

AR **-** Aspect Ratio

EAR **-** Eye Aspect Ratio

MAR **-** Mouth Aspect Ratio

IV. METHODOLOGY

The proposed system works as follows: A complete procedure is presented that moves the mouse from one place to another on the desktop through users’ face gestures. Initially, real-time user input is acquired from the user and the input is used for the functioning of the cursor. Hence, using this user input various operations can be performed.

The system design of the application is as follows:

Fig. 1 represents the algorithm used for face detection and the movement of the cursor. Initially the input is taken and preprocessing of the data is done to make it suitable for the application. Then the face is detected and points are marked on the face to trigger movement. If the mark on the face is displaced then the cursor is moved in accordance till it reaches the desired location. Then the click functions are checked if triggered then actions are performed else the system waits.

The system is in a constant loop where it waits for any displacement or and gestures given by the eye to perform the desired action.

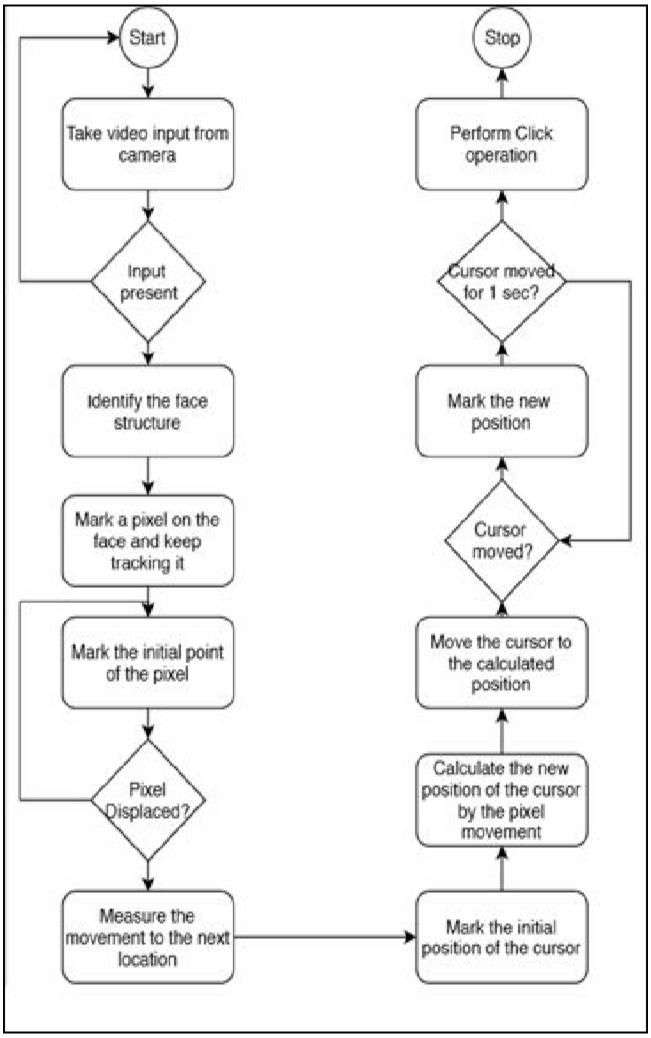

Fig. 1: System design

A. *User input*

The user has to place the laptop/desktop screen in close proximity to the face about half a meter. The webcam is used to capture the user's face which will be used further as input for cursor movements. A real-time video input will be taken by the webcam. A significant requirement of the user input should be good lighting conditions. The initial user input will be in RGB format.

B. *Preprocessing of the user input*

As shown in Fig.2, The input from the user will be initially flipped by 180 degrees using *cv2.flip( )* function wherein the left eye becomes the right eye and the right eye becomes a left eye. Further the video will be resized into a frame and converted into Grayscale using *cv2.cvtColor()* function [8].

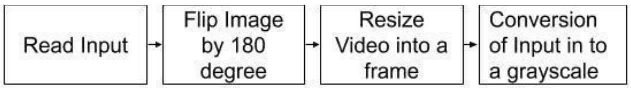

Fig. 2: Preprocessing of user input

C. *Face detection*

The faces are then detected in a grayscale frame and facial coordinates are located on the user's face. These facial landmark coordinates are converted into *NumPy* arrays. Further, the Left and Right eye coordinates are extracted and the eye aspect ratio is computed. The Euclidean distance between the vertical and horizontal mouth landmarks are computed and the mouth aspect ratio is obtained. The nose point is computed using the array values that are graphed on the user's face using *dlib* and the anchor point is the origin point through which other points are computed.

Further, mouth aspect ratio is computed using *np.ling.norm()* function. The horizontal and vertical mouth landmarks are located and mouth aspect ratio *(MAR)* is computed as,

$$MAR= \frac{A+B+C}{2*D} \tag{1}$$

where A, B, C are vertical landmarks.
and D is the horizontal landmark.

D. *Cursor functions:*

Various cursor functions like the input mode, single left click, single right-click, double click, Scroll and Drag have been implemented. The input mode can be enabled as well as disabled when not in use. The cursor moves left, right, up and down as per the movement of users' anchor point. If the mouth aspect ratio is greater than the threshold value the input mode is enabled and user input will be accepted. But if the mouth aspect ratio is less than the threshold value the input mode will be disabled and user input will not be accepted.

1. Input mode:

If the mouth aspect ratio is greater than the Mouth threshold value the input is accepted by the system and the input mode is enabled else the input mode is disabled and the input is not accepted by the system.

The movement of the cursor is defined by a function as,

function(*np*,*ap*,*w*,*h*,*m*):
if np > x + m * w:
then move right
else if nx < x - m * w:
then move left
else if ny > y + m * h:
then move down

```
else if  ny < y  -  m * h:
        then move up
    else return 0
```

where,

‘np’ is a nose point
‘nx’ and *‘ny’* are horizontal and vertical points of *‘np’*
‘ap’ is an anchor point
‘*x*’ and *‘y’* are horizontal and vertical point of *‘ap’*
‘w’ is width
‘h’ is height
‘m’ is the variable initialized to 1

1. Single-click:

When the aspect ratio of the left eye becomes less than the aspect ratio of the right eye, then the left click is implemented. Similarly, if the aspect ratio of the right eye is smaller than the aspect ratio of the left eye then right-click is implemented. *Clickstate* is the variable used to change the mode of clicks , the values stored in clickstate indicate different types of operations to be performed (i.e. single click , double click, drag). *MouseCount* is a counter used in drag mode.

a. Left-click function:

```
if left_EAR < right_EAR:
    if clickstate == 0:
        then single_left_click
    else if clickstate == 1:
        then double_left_click
    else if clickstate == 2:
        then drag_using_ left_click
```

In clickstate == 2 there are two different subparts, click and release.

```
i. Click and Release
    if mousecount == 0:
        press left button
    else if mousecount == 1:
        release left button
```

b. Right-click function:

```
if left_EAR > right_EAR:
    then single click
else
    return value 0;
```

2. Scroll mode:

If the aspect ratio of both the left and right eye together is less than the threshold value, the scroll mode is implemented. As the cursor reaches the upper and lower threshold value the cursor moves up and down respectively.

In scroll mode we take the mean of the left eye aspect ratio and right eye aspect ratio and this mean is stored in *ear*.

```
if ear <= Eye_ar_thresholdvalue, then eye counter increment to 1
if eye counter > eye_AR_frame:
then Scroll_Mode_Enabled
```

3. Double click mode:

If the click state value is 1 it performs a double click function.

```
else if clickstate == 1:
    then double_left_click is enabled
```

4. Drag mode:

If the click state value is 2 it performs a double click function.

```
else if clickstate == 2:
    then drag_using_leftclick is enabled
```

V. RESULTS

The proposed system is able to track faces and locate the cursor on the computer’s screen successfully. The user is able to track the mouse and control the movement of the mouse as well as various mouse functions like clicking, dragging, scrolling using different eye movements. This system is tested on multiple users, out of which two are physically disabled and one is physically fit. The respective snapshots of test cases are as shown below.

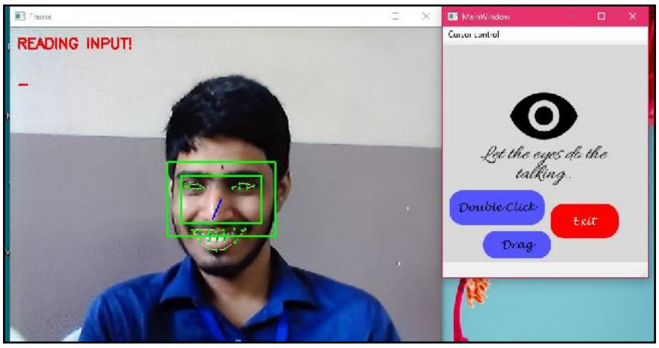

Fig. 3: Reading input

Fig. 3 shows input reading feature by the application. Input function becomes enabled when the user opens his mouth. The two boxes in the above figure represent the speed of the cursor movement. Once the control moves beyond the outer box the speed increases and when the control remains inside the inner box the speed lowers. The user's mouth is the only significant part required for obtaining input by the system application. The input function can also be disabled by again opening the mouth when the application is not in use.

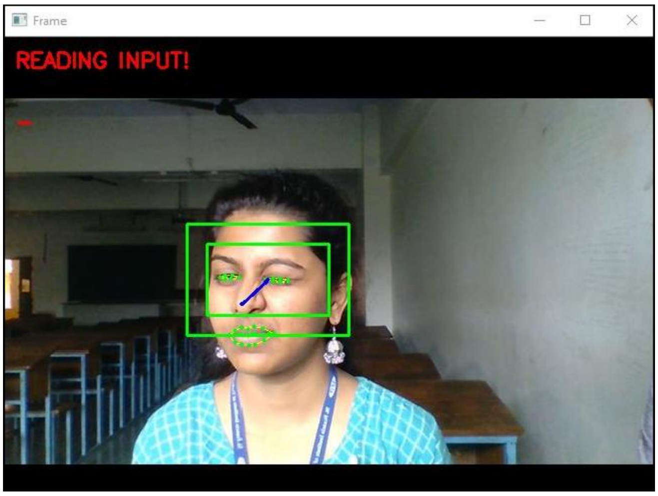

Fig. 4: Left click functionality

Fig. 4 shows the cursor moving towards the left and the left click functionality is implemented using the left eye. In click functionality, eyes are the significant part that are required for clicking the icons.

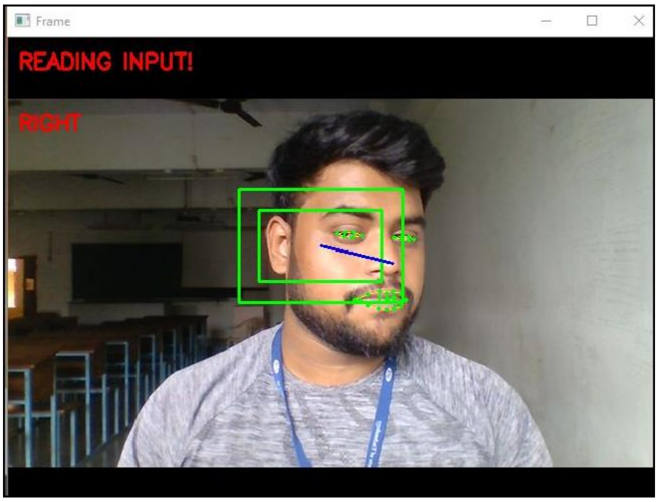

Fig. 5: Right click functionality

Fig. 5 shows the cursor moving towards the right and the right click functionality is implemented using the right eye. In click functionality, eyes are the significant part that are required for clicking the icons.

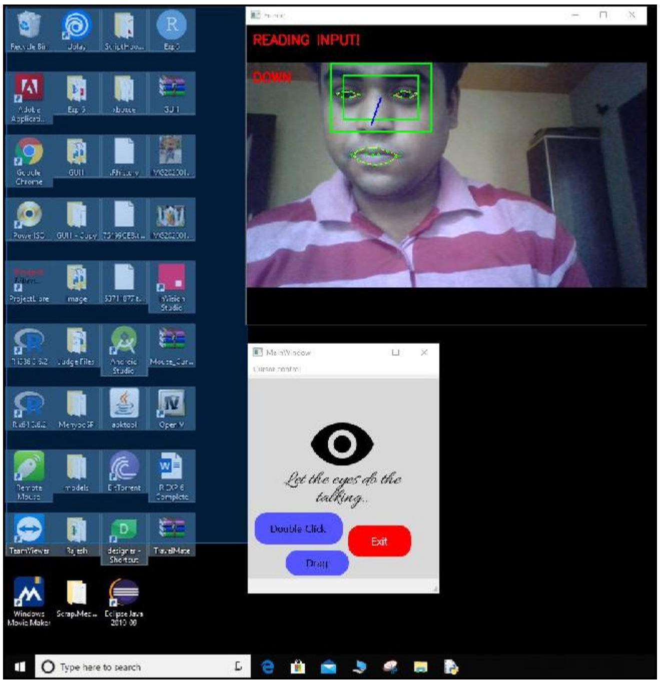

Fig. 6: Drag functionality

Fig. 6 shows the icons selected and further dragged over to other places. Drag function takes place when the user selects icon(s) using the left click that is by using the left eye and also releases using the left eye. A part of left click functionality is used in Drag.

A. *Test Cases*

The proposed application has been tested on 3 different users to check the feasibility and compatibility of the application. The system application was tested on users of various age groups. The feedback helped to improve the features and the application speed with respect to the use of the system. The results are as follows:

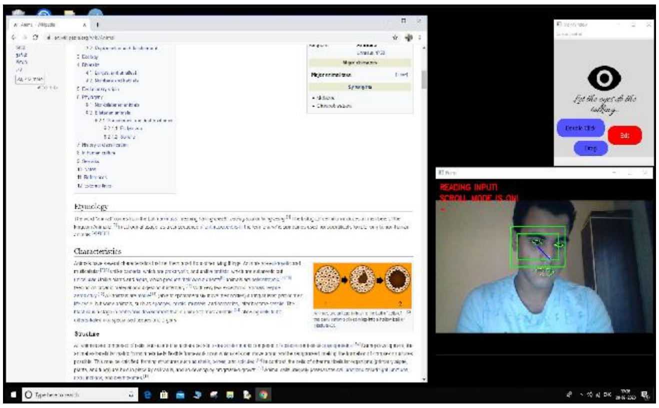

Fig. 7: User 1 (Age 18)

Fig. 7 shows a user of age 18. The user is physically fit and the user was compatible with the application and could use it perfectly. In the above figure, scroll functionality has been implemented by the user.

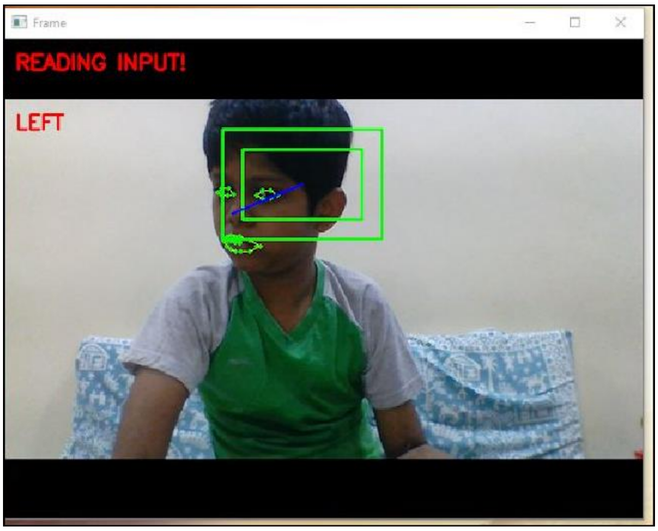

Fig. 8: User 2 (Age 13)

Fig. 8 shows a user of age 13. The user is physically challenged with hand deformation. The user was suitably able to use the system. In this Fig., the different operations of the cursor were performed by the user easily and also the face tracking of the user was performed smoothly.

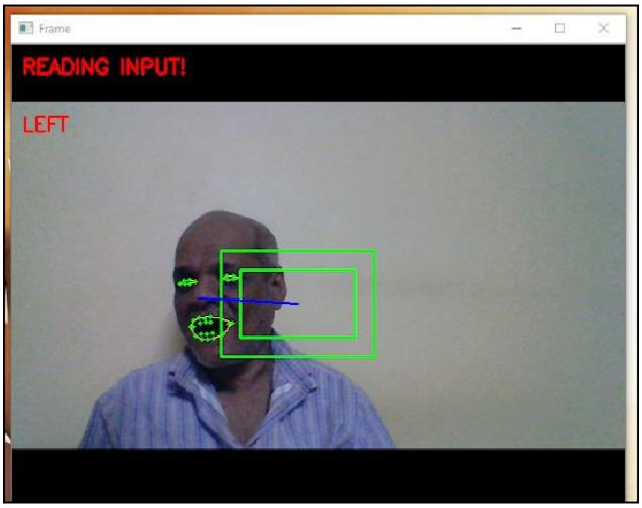

Fig. 9: User 3 (Age 57)

Fig. 9 shows a user of age 57. The user is physically challenged with 70% of disability. The user was able to perform cursor movements and cursor functions with ease.

VI. CONCLUSION

The proposed system would take real-time input from users with the help of OpenCV and run the application in the background. An anchor point on the user's face would control the cursor on the screen, opening of the mouth will calculate the distance between the upper and lower lips and will enable/disable the input. The left wink and the right wink would implement left and right-click functions respectively. Squinted eyes enable the scroll function in case of reading the documents or a file. It would thus facilitate the use and movement of the cursor through face gestures. This application can be implemented on laptops and desktops with inbuilt or external webcams. With the help of this application, the user can perform actions like moving the cursor in all directions, clicking functions, scroll function and drag function. Thus the system would be functional and useful for physically challenged users. One prominent advantage of the application is available in a ready-to-install executable file without the need of external packages. The application can be set to auto-start on system boot to avoid the inconvenience of manually launching the application.

Some system limitations are low lighting conditions because the user's face will not be clearly detected under low light. The system can only operate on devices with an integrated webcam or the webcam must be connected to the device explicitly.

Future work will include efficient user input in low light conditions. Also, along with desktops and laptops, this application has the scope of being implemented on android devices. This system can be applicable for building user-friendly applications by which one can paint, chat and play virtual games.

REFERENCES

[1] Morimoto, Carlos, Dave Koons, Arnon Amir, and Myron Flickner. "Real-time detection of eyes and faces."In Workshop on Perceptual User Interfaces, pp. 117-120,1998.

[2] Viola, Paul, and Michael Jones. "Fast and robust classification using asymmetric adaboost and a detector cascade."In Advances in neural information processing systems, pp.1311-1318,2002.

[3] Ruddarraju, Ravikrishna, Antonio Haro , Kris Nagel, Quan T. Tran, Irfan A. Essa, Gregory Abowd, and Elizabeth D. Mynatt."Perceptual user interfaces using vision-based eye tracking." In Proceedings of the 5th international conference on Multimodal interfaces, vol.no.5, pp.227-233, 2003.

[4] Anuar, Ammar, et al. "OpenCV based real-time video processing using android smartphone. "International Journal of Computer Technology and Electronics Engineering ,vol.no.3, pp: 1-6, 2011.

[5] Wankhede, Shrunkhala Satish, S. Chhabria, and R.V. Dharaskar. "Controlling mouse cursor using eye movement." International Journal of Application or Innovation in Engineering & Management 36, pp. 1-7, 2013.

[6] He, J., Roberson, S., Fields, B., Peng, J., Cielocha, S. and Coltea, J. “Fatigue detection using smartphones”. Journal of Ergonomics, vol.no.3(03), pp.1-7, 2013.

[7] Dey, Sanjay, et al. "Real Time Driver Fatigue Detection Based on Facial Behaviour along with Machine Learning Approaches”. IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON), pp. 135-140, 2019.

[8] Hu, Yuxiao, Longbin Chen, Yi Zhou, and Hongjiang Zhang. "Estimating face pose by facial asymmetry and geometry." In Sixth IEEE International Conference on Automatic Face and Gesture Recognition,vol.no.6, pp. 651-656, 2004.