Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

# HANDS FREE MOUSE CONTROL FOR PARAPLEGICS AND PARALYTICS

J – Component for the course

CSE4015 – Human Computer Interaction

**By**

Akshita Kapur

Advaith Chandra Srivastav

Ajit Singh

Riya Dasharath Gaonkar

**Under the Guidance of**

Prof. Deepa.K

Assistant Professor Sr.

# DECLARATION BY THE CANDIDATE

We hereby declare that the project report entitled "HANDS FREE MOUSE CONTROL FOR PARAPLEGICS AND PARALYTICS" submitted by us to Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of B. Tech in CSE is a record of J- component of project work carried out by us under the guidance of Dr. Deepa K. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore Institute of Technology, Vellore.

Date: 25-11-2021

## ACKNOWLEDGEMENT

We would like to thank Dr Deepa K for her continued support and guidance through out this project. We would also like to thank her and Vellore Institute of Technology for providing us with the opportunity to work on this project and explore a domain that incites wisdom and knowledge in us. It was a real delight completing this project and this was really helpful in understanding the course work. We would also lie to thank our parents for giving us the resources required to complete this project.

## 1.1 TITLE :

Hands free mouse control for paraplegic and paralytic people.

## 1.2 AIM :

To make the use of computer vision and facial detection to make a prototype of hands free mouse control for paralytic and paraplegic people.

## 1.3 OBJECTIVE

Computers are the most popular gadgets among this generation. Their applications are countless. They are utilised for research, entertainment, and education. People who are paralysed or paraplegic cannot move any parts of their bodies below the waist. The disorder known as paraplegia, sometimes known as leg paralysis, is caused by the loss of muscular function and can affect the entire or a portion of the torso, the legs, and the pelvic organs. Following a spinal cord injury, paraplegia can develop. Damage to the spinal column's discs, ligaments, or vertebrae is the root cause.

Many people with spinal cord injuries can live active, independent lives thanks to therapy, medication, and medical equipment. Because it addresses a practical issue encountered by those who are physically disadvantaged in this way, this project on cursor movement for paraplegics and paralytics is considered appropriate for the project. In fact, the Google search bar auto-completes to a number of items when the phrase "how do paraplegics" is entered in, including "how do paraplegics use the restroom.". We must thus devise a practical way to enable such physically challenged folks to conveniently access such technology equipment.

We are using Computer Vision concepts to try and solve the functionality gap between existing systems and the differently abled people who might want to use said systems, whether it be for personal use, leisure, work, or even a communication device.

We will make use of Facial Detection to create a prototype of an application which allows users who don’t have the use of their arms, to be able to access their computers. This will be done by enabling the mouse cursor to act according to certain sequences and actions performed by the person using their face and head to move said cursor and click on things.

Since most modern operating systems have an inbuilt virtual keyboard, this also solves the problem of not being able to use the physical keyboard.

**2. LITERATURE SURVEY:**

| **Title and Name of Journal** | **Authors** | **Proposed Methodology** | **Survey** |
|---|---|---|---|
| 1.Real Time Eye Blink Detection Using Facial Landmarks<br>21st Computer Vision Winter Workshop | Tereza Soukupova' and Jan C | A real-time algorithm to detect eye blinks in a video sequence from a standard camera is proposed. Recent landmark detectors, trained on in-the wild datasets exhibit excellent robustness against a head orientation with respect to a camera, varying illumination and facial expressions. It shows that the landmarks are detected precisely enough to reliably estimate the level of the eye opening. | This paper gives an idea of how to take up with the blinking patterns, facial expressions and the lighting conditions.<br>One of the most important features as how to track the movement and blinking of eyes in real-time. The system in play here analyses the pattern and duration of the blinks, using them to provide input to the computer in the form of a mouse click. |
| 2. One millisecond face alignment with an ensemble of regression trees<br>2014 IEEE Conference on Computer Vision and Pattern Recognition | Vahid Kazemi and Josephine Sullivan | This paper addresses the problem of Face Alignment for a single image. We show how an ensemble of regression trees can be used to estimate the face's landmark positions directly from a sparse subset of pixel intensities, achieving | This paper analyses the effect of the quantity of training data on the accuracy of the predictions and explore the effect of data augmentation using synthesized data.<br>It shows how the |

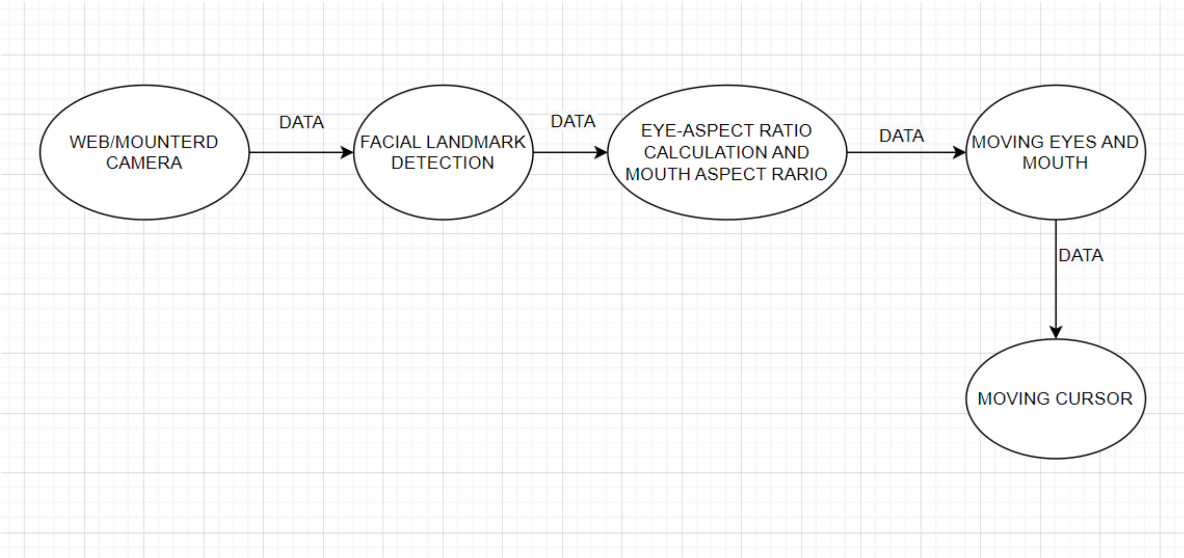
| | | | |
|---|---|---|---|
| | | super real-time performance with high quality predictions. We present a general framework based on gradient boosting for learning an ensemble of regression trees that optimizes the sum of square error loss and naturally handles missing or partially labeled data. | ensemble of regression trees can be used to estimate the face's landmark positions directly from a sparse subset of pixel Intensities achieving super real-time performance with high quality prediction. |
| 3. The first Facial Landmark localization<br>IEEE International Conference on Computer Vision Workshops | C Sagonas G. Tsimiropou Los, S. Zafeiriou and M. Pantic | Automatic facial point detection plays arguably the most important role in face analysis. Several methods have been proposed which reported their results on databases of both constrained and unconstrained conditions. Most of these databases provide annotations with different mark-ups and in some cases the are problems related to the accuracy of the fiducial points. The aforementioned issues as well as the lack of a evaluation protocol makes it difficult to compare performance between different systems. | This paper presents a challenge on showing how the facial point detection plays the most important role in face analysis. |

| | | | |
|---|---|---|---|
| 4. Real Time Eye Tracking and Blink Detection with USB Cameras<br><br>IEEE International Conference on Computer Vision Workshops | C Sagonas G. Tsimiropou Los, S. Zafeiriou and M. Pantic | A human-computer interface (HCI) system designed for use by people with severe disabilities is presented. People that are severely paralyzed or afflicted with diseases such as ALS (Lou Gehrig's disease) or multiple sclerosis are unable to move or control any parts of their bodies except for their eyes. The system presented here detects the user's eye blinks and analyses the pattern and duration of the blinks, using them to provide input to the computer in the form of a mouse click. | One of the most important features as how to track the movement and blinking of eyes in real-time.<br><br>The system in play here analyses the pattern and duration of the blinks, using them to provide input to the computer in the form of a mouse click. |
| 5. Drowsy Driver Detection System Using Eye Blink Patterns<br><br>International Conference on Machine and Web Intelligence | Taner Danisman, Ioan Marius Bilasco, Ghaabane Djeraba, Nacim lhaddaden e | This paper presents an automatic drowsy driver monitoring and accident prevention system that is based on monitoring the changes in the eye blink duration. Our proposed method detects visual changes in eye locations using the proposed horizontal symmetry feature of the eyes. Our new method detects eye blinks via a standard webcam in real-time at | This paper talks about how to tackle if a person is drowsy. The proposed method detects visual changes in eye locations using the proposed horizontal symmetry feature of the eyes. |

| | | | |
|---|---|---|---|
| | | 110fps for a 320×240 system detects eye blinks with a 94% accuracy with a 1% false positive rate. | |
| 6. Learning technologies for people with disabilities<br><br>Current Advances in Digital Learning Technologies | Mohsen Laabidi, Mohamed Jemni, Leila Jemni Ben Ayed , HejerBen Brahim Amal, Ben Jemaa | In this paper, they covered basic concepts of e-accessibility, universal design and assistive technologies, with a special focus on accessible e-learning systems. Then, we will present recent research works conducted in our research Laboratory LaTICE toward the development of an accessible online learning environment for persons with disabilities from the design and specification step to the implementation. | Many researchers developed specific tools dealing with specific needs for people with disabilities. However, these tools do not allow the user to adapt the contents to other needs.<br>- IMS/ISO specification<br>- MoodleAcc+ gives to learners the possibility to specify and edit his preferences through its Learner Assistance Tool. |
| 7. Eye-blink detection system for human–computer interaction<br><br>Spingerlink | Aleksandra Królak Pawel Strumillo | The system, capable of processing a sequence of face images of small resolution (320 x 240 pixels) with the speed of approximately 30 fps, is built from off the shelf components: a consumer-grade PC or a laptop and a | Separate short eye-blinks are assumed to be spontaneous and are not included in the designed eye-blink code. |

| | | | |
|---|---|---|---|
| | | medium quality webcam. The proposed algorithm allows for eye-blink detection, estimation of the eye-blink duration and interpretation of a sequence of blinks in real time to control a non-intrusive human-computer interface. | |
| 8. Smart Assistive Device for Physically Challenged People<br>NCESC – 2018 (Volume 6 – Issue 13 | Veekshita R R, Meghana R, Varsha Iyengar G, Thejaswini B R, Latha M | The main objective of this project is to provide a keypad device for the disabled people who cannot talk.<br>Arduino is an open source computer hardware and software company, project and use community that designs and manufactures single board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. Arduino board designs use a variety of microprocessor and controller. | The developed device will provide voice assistance to the bed-ridden individuals / aged people in their day-to-day operation of the applications present in their surroundings thus increasing the level of comfort. This project describes the design and working of a system which is useful for dumb people to communicate with one another and with normal people. This device forms a bridge between the physically disabled people and the normal people. |

## 3. ARCHITECTURAL DIAGRAM




## 4.1. METHODS AND DESCRIPTION ABOUT THE DIAGRAM

The project is centred around using facial landmarks to control the cursor of the mouse. The idea was to use existing facial landmark models to create a system which functions well and requires no external sensors or gadgets as explained in [1]. The primary function is to create a method for hands-free movement of the cursor, thus enabling the differently challenged to be able to use the computer for basic needs.

By using specific actions or expressions, we give commands to the computer to move the cursor. Various actions elicit various responses:

- Mouth Open - Activate/Deactivate Mouse Control
- Right Eye Wink - Right Click
- Left Eye Wink - Left Click
- Squinting - Activate and Deactivate scrolling
- Head Movement - Cursor movement or Scroll Movement

As mentioned, the project is based on detecting facial landmarks. A lot of things can be accomplished using these landmarks. We are, however, focusing on the changes in these landmarks as triggers for our application. For the purpose of detection, we are using a pre-built dlib library, which not only does quick face detection, but also 68 points on the x-y plane of the face, which comes in very handy.

By examining the above diagram, we can see that by using standard Python indexing, one can recognize facial features. One can extract these features and make use of it using the dlib library to detect along with OpenCV (real-time Computer Vision library), imutils (Python library) and code in Python 3.6 and above.

In terms of blink detection, only two sets of facial structures matter - the eyes. Each eye is represented by 6 x-y coordinates from p1 to p6 starting from the left corner of the eye and working its way around clockwise.

On the same principle as EAR, a Mouth Aspect Ratio (MAR) can be calculated, which will function the same way for all intents and purposes. Instead of 6 points, here 8 points are used. Using these predicted landmarks of the face, we can build appropriate features that will further allow us to detect certain actions, like using the eye-aspect-ratio (more on this below) to detect a blink or a wink, using the mouth-aspect-ratio to detect a yawn etc or maybe even a pout. In this project, these actions are programmed as triggers to control the mouse cursor. PyAutoGUI library was used to move the cursor around.

## 4.2 IMPLEMENTATION

## CODE:

```
from imutils import face_utils

from utils import *

import imutils

import numpy as np

import pyautogui as pag

import dlib

import cv2

# Thresholds and consecutive frame length for triggering the mouse action.

MOUTH_AR_THRESH = 0.3

MOUTH_AR_CONSECUTIVE_FRAMES = 5

EYE_AR_THRESH = 0.20

EYE_AR_CONSECUTIVE_FRAMES = 5

WINK_AR_DIFF_THRESH = 0.001

WINK_AR_CLOSE_THRESH = 0.20

WINK_CONSECUTIVE_FRAMES = 4

# Initialize the frame counters for each action as well as

# booleans used to indicate if action is performed or not

MOUTH_COUNTER = 0

EYE_COUNTER = 0

WINK_COUNTER = 0

INPUT_MODE = False
```

```
EYE_CLICK = False
LEFT_WINK = False
RIGHT_WINK = False
SCROLL_MODE = False
ANCHOR_POINT = (0, 0)
WHITE_COLOR = (255, 255, 255)
YELLOW_COLOR = (0, 255, 255)
RED_COLOR = (0, 0, 255)
GREEN_COLOR = (0, 255, 0)
BLUE_COLOR = (255, 0, 0)
BLACK_COLOR = (0, 0, 0)
# Initialize Dlib's face detector (HOG-based) and then create
# the facial landmark predictor
shape_predictor = "model/shape_predictor_68_face_landmarks.dat"
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(shape_predictor)
# Grab the indexes of the facial landmarks for the left and
# right eye, nose and mouth respectively
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
(nStart, nEnd) = face_utils.FACIAL_LANDMARKS_IDXS["nose"]
(mStart, mEnd) = face_utils.FACIAL_LANDMARKS_IDXS["mouth"]
# Video capture
vid = cv2.VideoCapture(0)
resolution_w = 1366
resolution_h = 768
cam_w = 640
cam_h = 480
unit_w = resolution_w / cam_w
unit_h = resolution_h / cam_h
while True:
```

```
# Grab the frame from the threaded video file stream, resize
# it, and convert it to grayscale
# channels
_, frame = vid.read()
frame = cv2.flip(frame, 1)
frame = imutils.resize(frame, width=cam_w, height=cam_h)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
# Detect faces in the grayscale frame
rects = detector(gray, 0)
# Loop over the face detections
if len(rects) > 0:
rect = rects[0]
else:
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
continue
# Determine the facial landmarks for the face region, then
# convert the facial landmark (x, y)-coordinates to a NumPy
# array
shape = predictor(gray, rect)
shape = face_utils.shape_to_np(shape)
# Extract the left and right eye coordinates, then use the
# coordinates to compute the eye aspect ratio for both eyes
mouth = shape[mStart:mEnd]
leftEye = shape[lStart:lEnd]
rightEye = shape[rStart:rEnd]
nose = shape[nStart:nEnd]
# Because I flipped the frame, left is right, right is left.
temp = leftEye
leftEye = rightEye
rightEye = temp
```

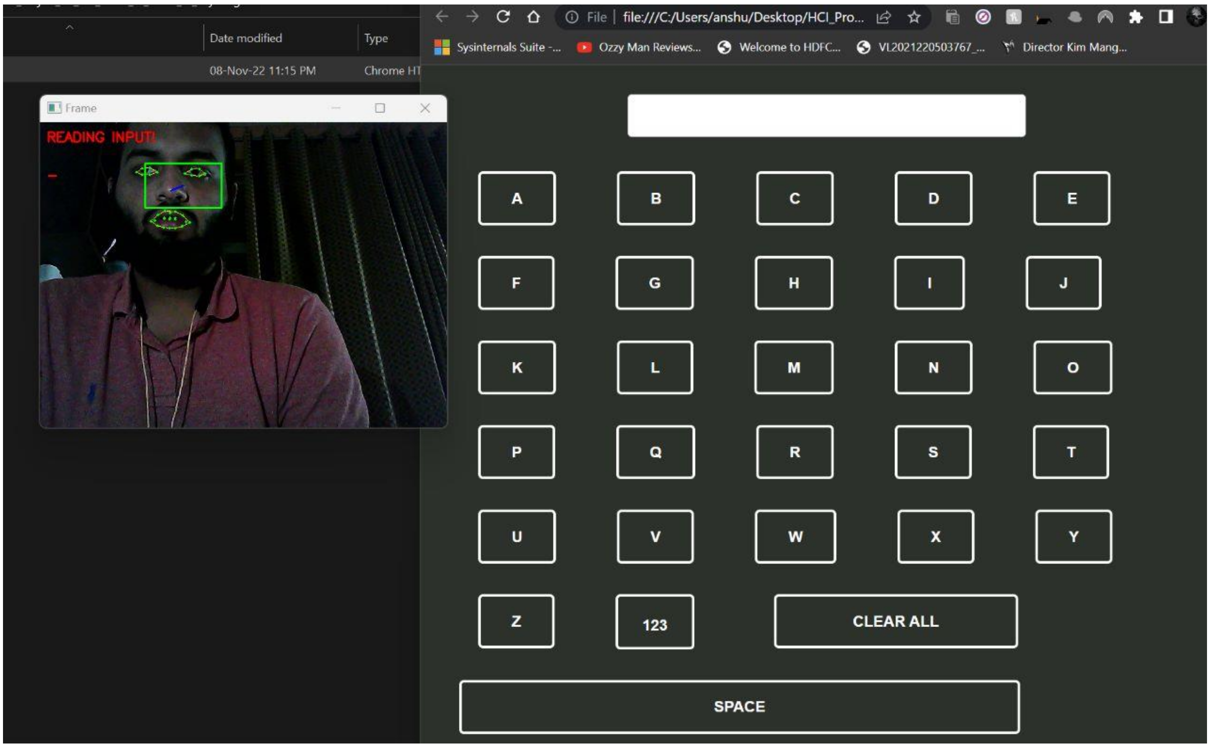
```
# Average the mouth aspect ratio together for both eyes
mar = mouth_aspect_ratio(mouth)
leftEAR = eye_aspect_ratio(leftEye)
rightEAR = eye_aspect_ratio(rightEye)
ear = (leftEAR + rightEAR) / 2.0
diff_ear = np.abs(leftEAR - rightEAR)
nose_point = (nose[3, 0], nose[3, 1])
# Compute the convex hull for the left and right eye, then
# visualize each of the eyes
mouthHull = cv2.convexHull(mouth)
leftEyeHull = cv2.convexHull(leftEye)
rightEyeHull = cv2.convexHull(rightEye)
cv2.drawContours(frame, [mouthHull], -1, YELLOW_COLOR, 1)
cv2.drawContours(frame, [leftEyeHull], -1, YELLOW_COLOR, 1)
cv2.drawContours(frame, [rightEyeHull], -1, YELLOW_COLOR, 1)
for (x, y) in np.concatenate((mouth, leftEye, rightEye), axis=0):
cv2.circle(frame, (x, y), 2, GREEN_COLOR, -1)
# Check to see if the eye aspect ratio is below the blink
# threshold, and if so, increment the blink frame counter
if diff_ear > WINK_AR_DIFF_THRESH:
if leftEAR < rightEAR:
if leftEAR < EYE_AR_THRESH:
WINK_COUNTER += 1
if WINK_COUNTER > WINK_CONSECUTIVE_FRAMES:
pag.click(button='left')
WINK_COUNTER = 0
elif leftEAR > rightEAR:
if rightEAR < EYE_AR_THRESH:
WINK_COUNTER += 1
if WINK_COUNTER > WINK_CONSECUTIVE_FRAMES:
pag.click(button='right')
```

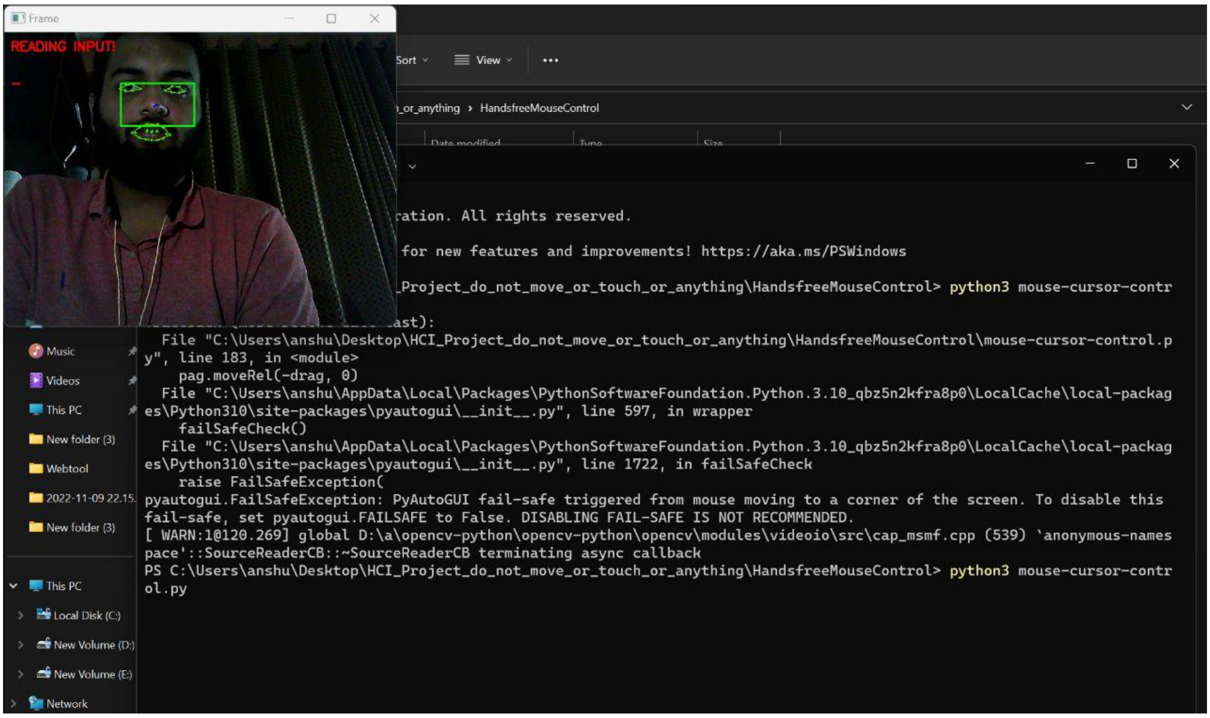
```
WINK_COUNTER = 0
else:
WINK_COUNTER = 0
else:
if ear <= EYE_AR_THRESH:
EYE_COUNTER += 1
if EYE_COUNTER > EYE_AR_CONSECUTIVE_FRAMES:
SCROLL_MODE = not SCROLL_MODE
# INPUT_MODE = not INPUT_MODE
EYE_COUNTER = 0
# nose point to draw a bounding box around it
else:
EYE_COUNTER = 0
WINK_COUNTER = 0
if mar > MOUTH_AR_THRESH:
MOUTH_COUNTER += 1
if MOUTH_COUNTER >= MOUTH_AR_CONSECUTIVE_FRAMES:
# if the alarm is not on, turn it on
INPUT_MODE = not INPUT_MODE
# SCROLL_MODE = not SCROLL_MODE
MOUTH_COUNTER = 0
ANCHOR_POINT = nose_point
else:
MOUTH_COUNTER = 0
if INPUT_MODE:
cv2.putText(frame, "READING INPUT!", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, RED_COLOR, 2)
x, y = ANCHOR_POINT
nx, ny = nose_point
w, h = 60, 35
multiple = 1
```

```
cv2.rectangle(frame, (x - w, y - h), (x + w, y + h), GREEN_COLOR,
2)
cv2.line(frame, ANCHOR_POINT, nose_point, BLUE_COLOR, 2)
dir = direction(nose_point, ANCHOR_POINT, w, h)
cv2.putText(frame, dir.upper(), (10, 90), cv2.FONT_HERSHEY_SIMPLEX,
0.7, RED_COLOR, 2)
drag = 18
if dir == 'right':
pag.moveRel(drag, 0)
elif dir == 'left':
pag.moveRel(-drag, 0)
elif dir == 'up':
if SCROLL_MODE:
pag.scroll(40)
else:
pag.moveRel(0, -drag)
elif dir == 'down':
if SCROLL_MODE:
pag.scroll(-40)
else:
pag.moveRel(0, drag)
if SCROLL_MODE:
cv2.putText(frame, 'SCROLL MODE IS ON!', (10, 60),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, RED_COLOR, 2)
# cv2.putText(frame, "MAR: {:.2f}".format(mar), (500, 30),
# cv2.FONT_HERSHEY_SIMPLEX, 0.7, YELLOW_COLOR, 2)
# cv2.putText(frame, "Right EAR: {:.2f}".format(rightEAR), (460, 80),
# cv2.FONT_HERSHEY_SIMPLEX, 0.7, YELLOW_COLOR, 2)
# cv2.putText(frame, "Left EAR: {:.2f}".format(leftEAR), (460, 130),
# cv2.FONT_HERSHEY_SIMPLEX, 0.7, YELLOW_COLOR, 2)
# cv2.putText(frame, "Diff EAR: {:.2f}".format(np.abs(leftEAR -
```

```
rightEAR)), (460, 80),
 # cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
 # Show the frame
 cv2.imshow("Frame", frame)
 key = cv2.waitKey(1) & 0xFF
 # If the `Esc` key was pressed, break from the loop
 if key == 27:
 break
# Do a bit of cleanup
cv2.destroyAllWindows()
vid.release()
```

## 5. RESULTS AND DISCUSSION






The above system works just as well as expected. Average response time is not more than 15-20 milliseconds, in a system with normal specs and no external graphic card. This shows that the system is highly responsive and compatible, with negligible to no lag between input and successive output. Furthermore, the tkinter app makes it easier to access the system using a GUI interface, instead of a CLI interface. This means that it can be easily used by the technologically challenged. Paired with the webtool interface, this system prototype covers all the bases it aimed to cover.

This project uses existing technology of Computer Vision and Facial Detection to create a system which enables differently abled people to perform the same tasks that we do so easily. Not only does it solve a technical problem, but a social one as well. The novelty is that unlike existing solutions, this system does not require any extra hardware or attachments. That makes it highly portable and affordable. This prototype also involves a web-based interface and a tkinter app so that users can easily type, an additional feature to just moving the cursor.

## 6. CONCLUSION:

The lack of software to address the issues faced by those with disabilities will have a significant impact on the state of the world today as a result of this effort. Our programme makes it simple for many of these folks to use.

We have made an effort to design the software in a way that enables users to interact with computers without physically touching the mouse, relying instead on little facial motions. The main idea behind this is to use facial landmarks to move the mouse cursor.

We instruct the computer to move the pointer by utilising precise gestures or emotions. Mouth Open - Mouse Control Activate/Deactivate, Right Eye Wink - Mouse Control Left Click, Left Eye Wink - Mouse Control Right Click, Squinting - Mouse Control Activate and Deactivate

Cursor or scroll movement is head movement. The primary components of our project were these.

**7. REFERENCES:**


Soukupova, T., & Cech, J. (2016, February). Eye blink detection using facial landmarks. In *21st computer vision winter workshop, Rimske Toplice, Slovenia*.

Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1867-1874).

C. Sagonas, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, "300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge," 2013 IEEE International Conference on Computer Vision Workshops, 2013, pp. 397-403, doi: 10.1109/ICCVW.2013.59.

Chau, Michael & Betke, Margrit. (2005). Real Time Eye Tracking and Blink Detection with USB Cameras. Boston Univ. Comput. Sci.. 2215.

Danisman, T., Bilasco, I. M., Djeraba, C., & Ihaddadene, N. (2010, October). Drowsy driver detection system using eye blink patterns. In *2010 International Conference on Machine and Web Intelligence* (pp. 230-233). IEEE.

Laabidi, M., Jemni, M., Ayed, L. J. B., Brahim, H. B., & Jemaa, A. B. (2014). Learning technologies for people with disabilities. *Journal of King Saud University-Computer and Information Sciences*, *26*(1), 29-45.

Królak, A., & Strumiłło, P. (2012). Eye-blink detection system for human–computer interaction. *Universal Access in the Information Society*, *11*(4), 409-419.

Veekshita, R. R., Meghana, R., Iyengar, G. V., Thejaswini, B. R., & Latha, M. (2018). Smart Assistive Device for Physically Challenged People. *Int. Jour. of Engineering Research & Technology*, *6*(13).