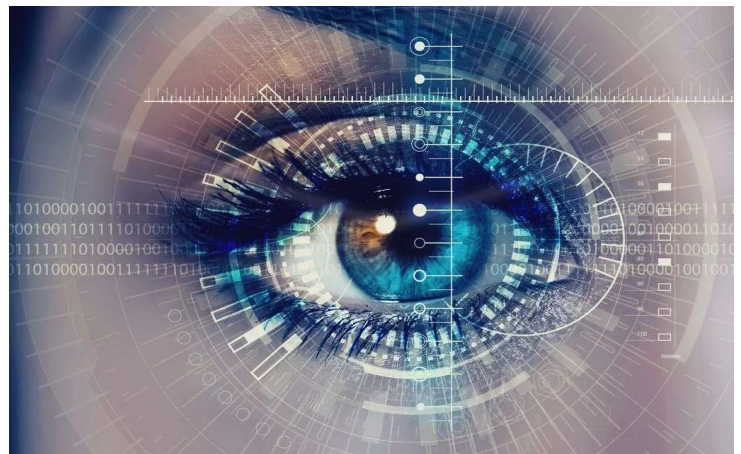


Project Report Curser Movement Using Eyes

Atharva Gadekar
21111015

Tejas Sinkar
21111047



- **Project Name:** Cursor Movement Using Eyes
- **Atharva Gadekar-21111015**
- **Tejas Sinkar-21111047**

- **Project Aim:** To move the cursor using face and eyes.

- **Objectives:**

- We move the cursor using eyes and face
- This project is totally based on python language.
- In this project we have used some python libraries
- We can click the mouse cursor with blink of an eye.

- **Need & Relevance:**

We can easily move the cursor with the help of human eyes.

We can click any button using just blinking of an eye.

- Controlling the mouse by a physically challenged person is really a tough one. To find a solution for the people who cannot use the Mouse physically, we have proposed this mouse cursor control using Eye Movements. This is an alternative way of accessing a computer using eye movements to control the mouse. For someone who finds touchscreens, mouse inaccessible, this is an alternative method to allow a user to operate their computer, using the movement of their eyes. Eye movement can be regarded as a pivotal real-time input medium for human-computer communication, which is especially important for people with physical disability. In order to improve the reliability, mobility, and usability of eye tracking technique in user-computer dialogue, a novel eye control system is proposed in this system using Webcam and without using any extra hardware. The proposed system focuses on providing a simple and convenient interactive mode by only using user's eye. The usage flow of the proposed system is designed to perfectly follow human natural habits. This movement of cursor according to eyes and face position which can be used to control the cursor on the screen using webcam and implemented using Python.

- **Expected Outcome:** we can move the cursor easily with our eyes and face.

- **References:**

- https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html
- https://google.github.io/mediapipe/solutions/face_mesh
- <https://pypi.org/project/PyAutoGUI/>

- 1.Code for opening the camera

- `import cv2`

`'used for image processing' cam =`

`cv2.VideoCapture(0)`

`'image processing and read the camera through
CV2 package'`

`'0 is an index number' while True:`

`'''in this project the video is running
every frame after`

`frame so here we use the while loop and this loop
is running forever '''`

`frame, frame = cam.read()`

`'to read the camera and get the frame of
every video'`

`'cam.read() It will call the camera to
CV2 to read the camera whatever is running
from the camera'`

`cv2.imshow('Cursor Movement Using Eyes',
frame)`

`'imshow: to show some image into the
camera '`

`cv2.waitKey(1)`

`'wait key for 1 second'`

-

2.Code for facedetection

```
import cv2
'used for image processing'

import mediapipe
'media pipe is used for the detect the face and eventually
it detects the eyes too like winking and movement of
eyes'

cam = cv2.VideoCapture(0)
'image processing and read the camera through CV2
package'
face_mesh = mediapipe.solutions.face_mesh.FaceMesh()
while True:
'''in this project the video is running every frame after
frame so here we use the while loop and this loop is
running forever '''

    _, frame = cam.read()
    '''to read the camera and get the frame of every
    video
    cam.read() It will call the camera to CV2 to read
the camera whatever is running from the camera'''
    rgb_frame = cv2.cvtColor(frame , cv2.COLOR_BGR2RGB)
    '''to detect the face and whenever you detecting the
face
'the face in greyscale or in an any colour''' output =
    face_mesh.process(rgb_frame) hotspot_points =
    output.multi_face_landmarks
    '''it will detect the hotspot points on your
    face and after recognizing it using that points
whether it the the actual face or not'''
    print(hotspot_points)
    cv2.imshow('Cursor Movement Using Eyes', frame)
    'imshow: to show some image into the camera '
    cv2.waitKey(1)
    'wait key for 1 second'
```

3.Code for landmarks on Face

```
import cv2
'used for image processing'

import mediapipe
'media pipe is used for the detect the face and eventually it detects the eyes too like winking and movement of eyes'

cam = cv2.VideoCapture(0)
'image processing and read the camera through CV2 package'
face_mesh = mediapipe.solutions.face_mesh.FaceMesh(refine_landmarks=True) 'to get much refine landmarks'
'''this refine_landmarks parameter have total 478 landmarks on the face and every landmark identifies different part of the face'''
while True:
    '''in this project the video is running every frame after frame so here we use the while loop and this loop is running forever'''
    __, frame = cam.read()
    '''to read the camera and get the frame of every video
    cam.read() It will call the camera to CV2 to read the camera whatever is running from the camera'''
    frame = cv2.flip(frame,1)
    'this means you are flipping vertically '
    rgb_frame = cv2.cvtColor(frame , cv2.COLOR_BGR2RGB) '''to detect the face and whenever you detecting the face 'the face in greyscale or in an any colour'''
    output = face_mesh.process(rgb_frame) landmark_points =
    output.multi_face_landmarks '''it will detect the hotspot points on your face and after recognizing it using that points whether it the the actual face or not.
    many.face = we will detecting for many faces ''' if landmark_points:
    landmarks = landmark_points[0].landmark frame_height , frame_width , _ =
    frame.shape for landmark in landmarks[474:478]:
    'loop for every hotspot on the face and draw the loop between them' '470:478 this range of index is for detect the iris'
    x = int(landmark.x * frame_width)
    'there are three coordinates x-axis for width , y-axis for height and z for distance between camera and the person '
    y = int(landmark.y * frame_height) cv2.circle(frame ,(x,y), 3 ,(0,255,0))
    'Frame for where to draw the circle ' '(x,y) is a centre'
    '3 is for radius of a circle'
    '0 is Red, 255 for green, 0 is blue' print(x , y)
    'it will show some fraction numbers and that numbers shows the position on the screen'
    cv2.imshow('Cursor Movement Using Eyes', frame) 'imshow: to show some image into the camera ' cv2.waitKey(1)
    'wait key for 1 second'
```

```

4.Code for movement of mouse import cv2
'used for image processing'

import mediapipe
'media pipe is used for the detect the face and eventually it detects the eyes
too like winking and movement of eyes'

import pyautogui
cam = cv2.VideoCapture(0)
'image processing and read the camera through CV2 package'
face_mesh = mediapipe.solutions.face_mesh.FaceMesh(refine_landmarks=True) 'to
get much refine landmarks'
'''this refine_landmarks parameter have total 468 landmarks on the face and every
landmark identifies different part of the face'''
screen_width, screen_height = pyautogui.size()
'to capture the total screen size' while True:
'''in this project the video is running every frame after
frame so here we use the while loop and this loop is running forever
'''
_, frame = cam.read()
'''to read the camera and get the frame of every video
cam.read() It will call the camera to CV2 to read the camera whatever is running
from the camera'''
frame = cv2.flip(frame,1)
'this means you are flipping vertically '
rgb_frame = cv2.cvtColor(frame , cv2.COLOR_BGR2RGB) '''to detect the face and
whenever you detecting the face'the face in greyscale or in an any colour'''
output = face_mesh.process(rgb_frame)
landmark_points = output.multi_face_landmarks '''it will detect the hotspot
points on your
face and after recognizing it using that points whether it the the actual face or
not.
many.face = we will detecting for many faces '''
if landmark_points:
landmarks = landmark_points[0].landmark frame_height , frame_width, _ =
frame.shape
for id, landmark in enumerate(landmarks[470:478]):
'loop for every hotspot on the face and draw the loop between them' '470:478 this
range of index is for detect the iris'
'enumerate will give you the ID or the index and also give the landmark
on your face'
x = int(landmark.x * frame_width)
'there are three coordinates x-axis for width , y-axis for height and z for
distance between camera and the person '
y = int(landmark.y * frame_height)
cv2.circle(frame , (x, y), 3,(0,255,0)) 'Frame for where to draw the circle ' '3
is for radius of a circle'
'255 for green'
'print(x , y) it will show some fraction numbers and that numbers shows the
position on the screen'
if id == 1:
screen_x = screen_width / frame_height * x screen_y = screen_height /
frame_height * y pyautogui.moveTo(screen_x,screen_y)
cv2.imshow('Cursor Movement Using Eyes', frame) 'imshow: to show some image into
the camera 'cv2.waitKey(1)
'wait key for 1 second'

```


5.code for click operation

```
import cv2
'used for image processing'

import mediapipe
'media pipe is used for the detect the face and eventually it
detects the eyes too like winking and movement of eyes'

import pyautogui
cam = cv2.VideoCapture(0)
'image processing and read the camera through CV2 package' face_mesh
=mediapipe.solutions.face_mesh.FaceMesh(refine_landmarks=True)
'to get much refine landmarks'
'''this refine_landmarks parameter have total 478 landmarks on the
face and
every landmark identifies different part of the face'''
screen_width, screen_height = pyautogui.size() 'to capture the
total screen size'
while True:
'''in this project the video is running every frame after frame so
here we use the while loop and this loop is
running forever'''

    frame, frame = cam.read()
    '''to read the camera and get the frame of every video
    cam.read() It will call the camera to CV2 to read the camera
whatever is running from the camera''' frame =
    cv2.flip(frame,1)
'this means you are flipping vertically '
rgb_frame = cv2.cvtColor(frame , cv2.COLOR_BGR2RGB) '''to
detect the face and whenever you detecting the face 'the
face in greyscale or in an any colour'''
output = face_mesh.process(rgb_frame) landmark_points =
output.multi_face_landmarks
'''it will detect the hotspot points on your
face and after recognizing it using that points whether it the
the actual face or not.
many.face = we will detecting for many faces '''
if landmark_points:
landmarks = landmark_points[0].landmark frame_height ,
frame_width, _ = frame.shape
for id, landmark in enumerate(landmarks[474:478]):
'loop for every hotspot on the face and draw the loop between
them'
'470:478 this range of index is for detect the iris'
'enumerate will give you the ID or the index and also give the
landmark on your face'
x = int(landmark.x * frame_width)
'there are three coordinates x-axis for width , y-axis for
height and z for distance between camera and the person '
y = int(landmark.y * frame_height)
```

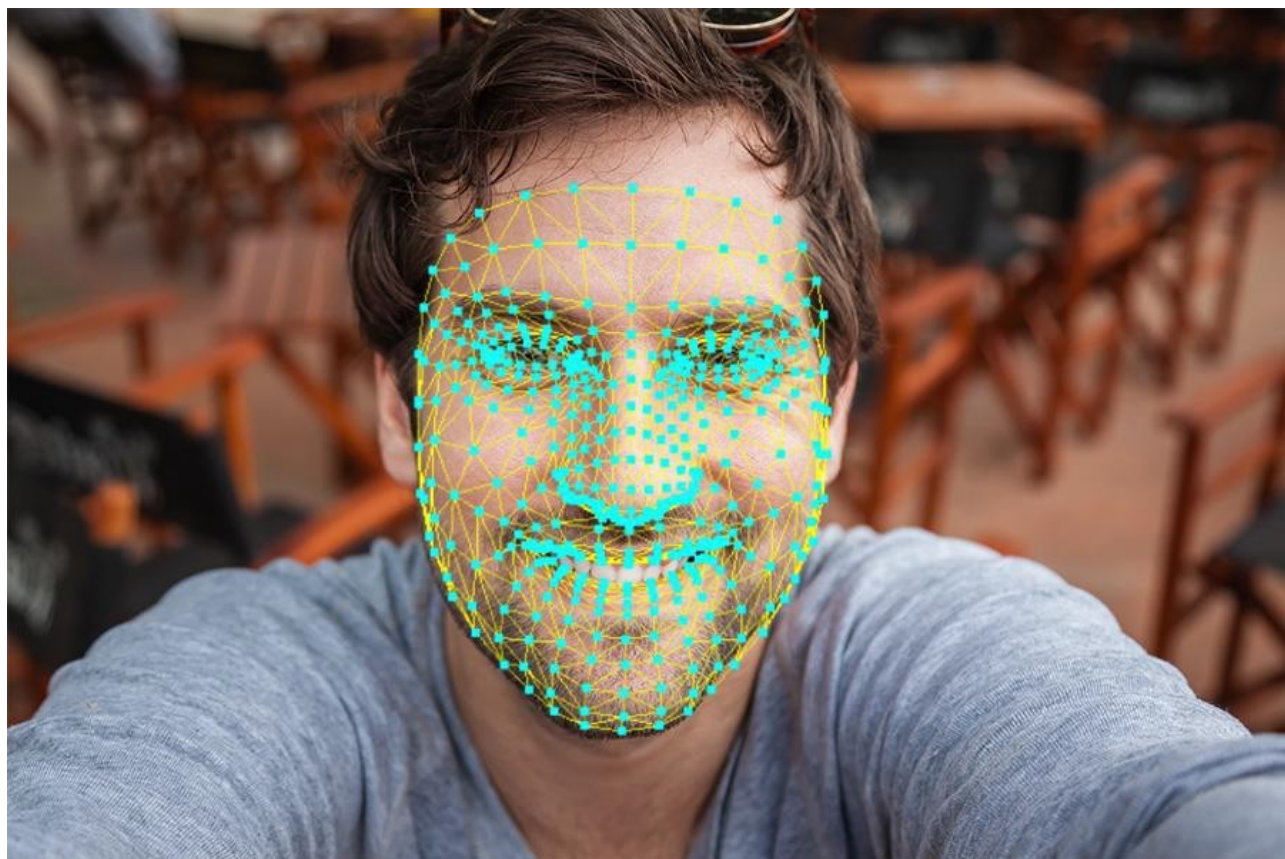
```

        cv2.circle(frame , (x, y), 3,(0,255,0) )
        'Frame for where to draw the circle ' of
        '3 is for radius a circle'
        '255 for green'
        'print(x , y) it will show some fraction numbers and
that numbers shows the position on the screen' if id == 1:
    screen_x = screen_width / frame_height * x
    screen_y = screen_height / frame_height * y
    pyautogui.moveTo(screen_x,screen_y)
    left_eye = [landmarks[145], landmarks[159]] '''these
        numbers are the landmarks of our eyes.
we can insert the array and loop through it''' for
    landmark in left_eye:

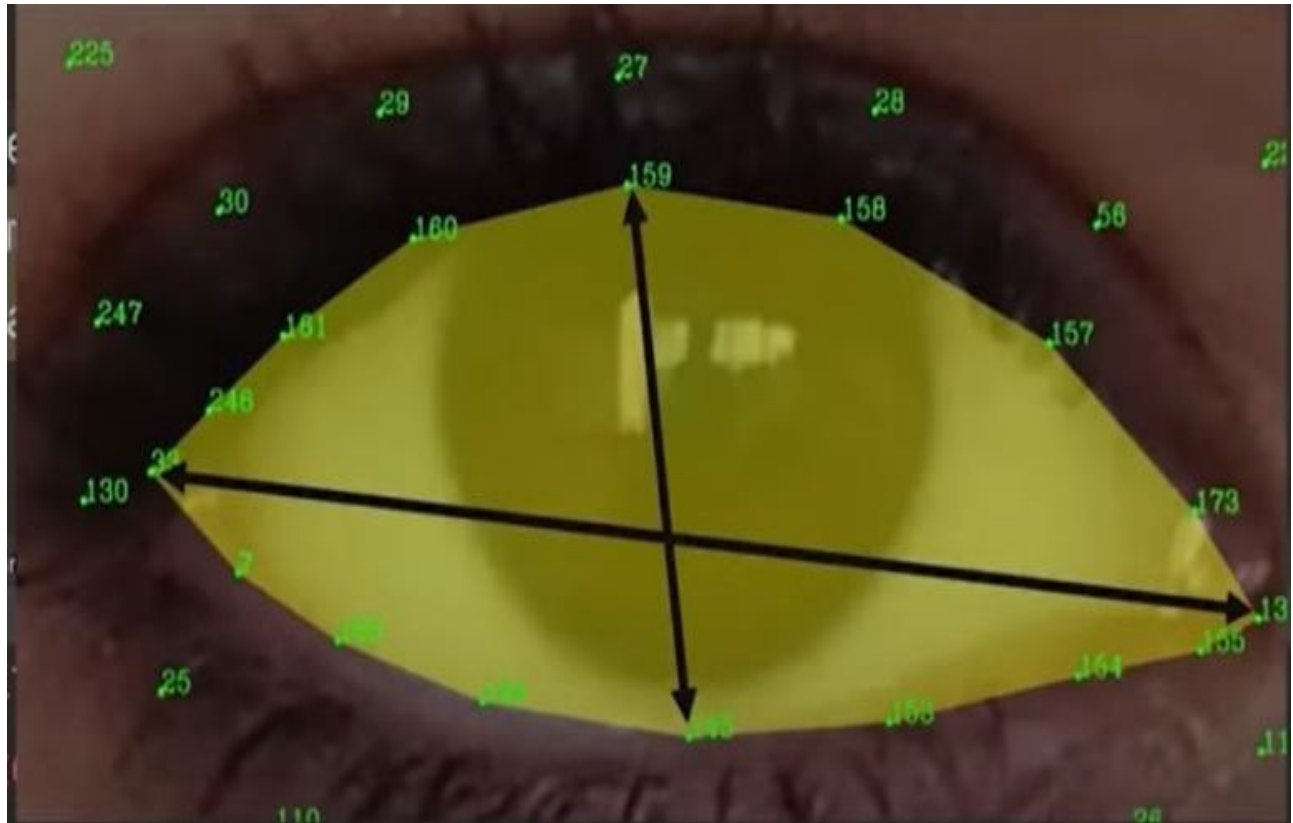
        x = int(landmark.x * frame_width)
        '''these lines are detecting the axis on our face'''
        y = int(landmark.y * frame_height)
        cv2.circle(frame , (x, y), 3,(0,255,255))
        'print(left_eye[0] left_eye[1].y)'
        if(left_eye[0].y - left_eye[1].y) < 0.005: print('Clicked')
        '''if the values are above of 0.021... then eyes are
        and
open    if that values are below of 0.001 then eyes are closed
'''
cv2.imshow('Cursor Movement Using Eyes', frame)
'imshow: to show some image into the camera '
cv2.waitKey(1)
'wait key for 1 second'

```

Landmarks on face!!!



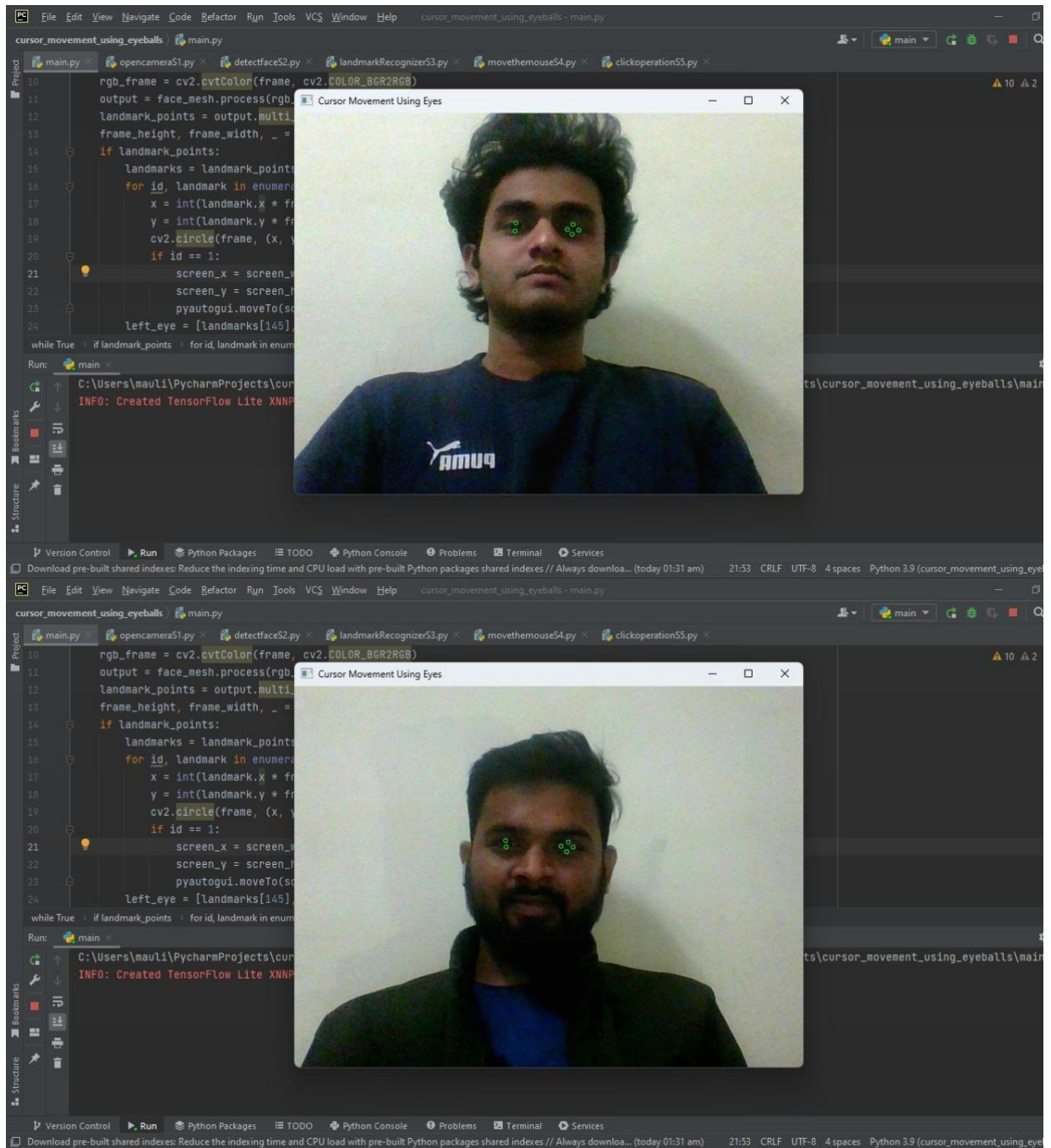
Index numbers of eyes



CODE FOR MOVEMENT OF MOUSE CURSOR USING EYE

```
import cv2
import mediapipe as mp import pyautogui
cam = cv2.VideoCapture(0) face_mesh =
mp.solutions.face_mesh.FaceMesh(refine_landmarks=
T
e)
screen_w, screen_h = pyautogui.size() while True:
_, frame = cam.read() frame = cv2.flip(frame, 1)
rgb_frame = cv2.cvtColor(frame,
cv2.COLOR_BGR2RGB)
output = face_mesh.process(rgb_frame)
landmark_points = output.multi_face_landmarks
frame_h, frame_w, _ = frame.shape
if landmark_points:
landmarks = landmark_points[0].landmark
for id, landmark in enumerate(landmarks[474:478]): x
= int(landmark.x * frame_w)
y = int(landmark.y * frame_h)
```

```
cv2.circle(frame, (x, y), 3, (0, 255, 0)) if id
== 1:
screen_x = screen_w * landmark.x
screen_y = screen_h * landmark.y
pyautogui.moveTo(screen_x, screen_y) left
= [landmarks[145], landmarks[159]] for
landmark in left:
x = int(landmark.x * frame_w) y =
int(landmark.y * frame_h)
cv2.circle(frame, (x, y), 3, (0, 255, 255))
if (left[0].y - left[1].y) < 0.004:
pyautogui.click() pyautogui.sleep(1)
cv2.imshow('Eye Controlled Mouse',
frame) cv2.waitKey(1)
```



Thank You...!



Any Questions...?

