**Detailed and Explained code**

**Library to install**

!pip install mediapipe

!pip install opencv-contrib-python

!pip install PyAutoGUI

**Actual code**

import cv2

import mediapipe as mp

import pyautogui

# Initialize the webcam

cam = cv2.VideoCapture(0)

# Initialize the face mesh model

face\_mesh = mp.solutions.face\_mesh.FaceMesh(refine\_landmarks=True)

# Get the screen size

screen\_w, screen\_h = pyautogui.size()

while True:

# Read a frame from the webcam

\_, frame = cam.read()

# Flip the frame horizontally to match the mirror view of the webcam

frame = cv2.flip(frame, 1)

# Convert the frame to RGB format

rgb\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

# Process the frame with the face mesh model

output = face\_mesh.process(rgb\_frame)

# Extract the facial landmarks from the output

landmark\_points = output.multi\_face\_landmarks

# Get the frame dimensions

frame\_h, frame\_w, \_ = frame.shape

# If any facial landmarks are detected

if landmark\_points:

# Extract the landmarks for the left and right eyes

landmarks = landmark\_points[0].landmark

for id in range(474, 478):

# Get the coordinates of the current landmark

x = int(landmarks[id].x \* frame\_w)

y = int(landmarks[id].y \* frame\_h)

# Draw a circle on the frame at the landmark's position

cv2.circle(frame, (x, y), 3, (0, 255, 0))

# If the current landmark corresponds to the center of the left eye

if id == 1:

# Calculate the corresponding screen coordinates

screen\_x = screen\_w \* landmarks[id].x

screen\_y = screen\_h \* landmarks[id].y

# Move the mouse to the calculated screen coordinates

pyautogui.moveTo(screen\_x, screen\_y)

# Extract the landmarks for the left eye

left = [landmarks[145], landmarks[159]]

# For each landmark in the left eye

for landmark in left:

# Get the coordinates of the current landmark

x = int(landmark.x \* frame\_w)

y = int(landmark.y \* frame\_h)

# Draw a circle on the frame at the landmark's position

cv2.circle(frame, (x, y), 3, (0, 255, 255))

# If the distance between the top and bottom landmarks of the left eye is less than a threshold

if (left[0].y - left[1].y) < 0.004:

# Click the left mouse button

pyautogui.click()

# Wait for a short period of time to avoid rapid clicking

pyautogui.sleep(1)

# Display the resulting frame

cv2.imshow('Eye Controlled Mouse', frame)

# Wait for a key press and exit the loop if the 'q' key is pressed

if cv2.waitKey(1) == ord('q'):

break

# Release the webcam and destroy all windows

cam.release()

cv2.destroyAllWindows()