Göz Takibi Mouse Kontrol Yazılımı

```
import cv2
import pyautogui
import numpy as np
from picamera2 import Picamera2

SCREEN_W, SCREEN_H = pyautogui.size()
SMOOTHING = 20
pyautogui.FAILSAFE = False

# NOT: username kısmına kendi raspberrypi hesabınızın ismini yazın !
face_cascade = cv2.CascadeClassifier(
    '/home/username/min1iforge3/envs/goz_env/lib/python3.11/site-packages/cv2/data/
haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier(
    '/home/username/miniforge3/envs/goz_env/lib/python3.11/site-packages/cv2/data/
haarcascade_eye.xml')

picam2 = Picamera2()
config = picam2.create_preview_configuration(
    main={"format": "XBGR8888", "size": (640, 480)})
picam2.configure(config)
picam2.start()

pos_history = []
blink_counter = 0
prev_eye_h = None

pyautogui.moveTo(SCREEN_W // 2, SCREEN_H // 2)
print("Kamera baslatildi, q ile cikis")

def find_pupil(eye_roi_gray):
    eye_roi_gray = cv2.GaussianBlur(eye_roi_gray, (7, 7), 0)
    _, thresh = cv2.threshold(eye_roi_gray, 40, 255, cv2.THRESH_BINARY_INV)
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    if contours:
        largest = max(contours, key=cv2.contourArea)
        M = cv2.moments(largest)
        if M["m00"] > 0:
            cx = int(M["m10"] / M["m00"])
            cy = int(M["m01"] / M["m00"])
            return cx, cy
    return None

while True:
    frame = picam2.capture_array()

    # XBGR8888 -> 4 kanal geliyor, ilk 3'ü al = BGR, renkler doğal
    frame = frame[:, :, :3]
    frame = cv2.flip(frame, 1)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    h, w = frame.shape[:2]

    faces = face_cascade.detectMultiScale(gray, 1.3, 5, minSize=(80, 80))

    for (fx, fy, fw, fh) in faces:
        cv2.rectangle(frame, (fx, fy), (fx+fw, fy+fh), (0, 255, 0), 1)
        roi_gray = gray[fy:fy+fh, fx:fx+fw]
        eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 10, minSize=(20, 20))
```

```
        if len(eyes) > 0:
            eyes = [e for e in eyes if e[1] < fh * 0.55]
            if len(eyes) == 0:
                continue

            eyes = sorted(eyes, key=lambda e: e[0])
            ex, ey, ew, eh = eyes[0]

            eye_roi_gray = roi_gray[ey:ey+eh, ex:ex+ew]
            pupil = find_pupil(eye_roi_gray)

            if pupil:
                px, py = pupil
                eye_cx = fx + ex + px
                eye_cy = fy + ey + py
            else:
                eye_cx = fx + ex + ew // 2
                eye_cy = fy + ey + eh // 2

            cv2.circle(frame, (eye_cx, eye_cy), 4, (0, 200, 255), -1)

            sx = int(np.interp(eye_cx, [w*0.38, w*0.62], [0, SCREEN_W]))
            sy = int(np.interp(eye_cy, [h*0.28, h*0.62], [0, SCREEN_H]))

            pos_history.append((sx, sy))
            if len(pos_history) > SMOOTHING:
                pos_history.pop(0)

            if len(pos_history) >= SMOOTHING:
                ax = int(np.mean([p[0] for p in pos_history]))
                ay = int(np.mean([p[1] for p in pos_history]))
                pyautogui.moveTo(ax, ay, duration=0)

            if prev_eye_h is not None:
                if eh < prev_eye_h * 0.75:
                    blink_counter += 1
                else:
                    if blink_counter >= 1:
                        pyautogui.click()
                        cv2.putText(frame, "TIKLANDI!", (20, 50),
                                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
                        blink_counter = 0
            prev_eye_h = eh

        break

    cv2.putText(frame, f"Yuz:{len(faces)}", (10, 30),
                cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 255, 0), 2)

    cv2.imshow("Goz Mouse | q:cikis", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

picam2.stop()
cv2.destroyAllWindows()
```