```python
from math import hypot

import cv2
import dlib

detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(r"C:\Users\rkssp\Downloads\facial-landmarks-recognition-master\facial-landmarks-recognition-master\shape_predictor_68_face_landmarks.dat")

cap = cv2.VideoCapture(0)

def midpoint(p1,p2):
    return int((p1.x + p2.x)/2), int((p1.y + p2.y)/2)

def get_blinking_ratio(eye_points,facial_landmarks):
    left_point = facial_landmarks.part(eye_points[0]).x, facial_landmarks.part(eye_points[0]).y
    right_point = facial_landmarks.part(eye_points[3]).x , facial_landmarks.part(eye_points[3]).y
    center_top = midpoint(facial_landmarks.part(eye_points[1]), facial_landmarks.part(eye_points[2]))
    center_bottom = midpoint(facial_landmarks.part(eye_points[5]), facial_landmarks.part(eye_points[4]))

    hor_line_lenght = hypot((left_point[0] - right_point[0]), (left_point[1] - right_point[1]))
    ver_line_lenght = hypot((center_top[0] - center_bottom[0]), (center_top[1] - center_bottom[1]))

    ratio = hor_line_lenght / ver_line_lenght
    return ratio

while True:
    success,img = cap.read()
    imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = detector(imgGray)

    for face in faces:
        landmarks = predictor(imgGray,face)
        left_eye_ratio = get_blinking_ratio([36, 37, 38, 39, 40, 41], landmarks)
        right_eye_ratio = get_blinking_ratio([42, 43, 44, 45, 46, 47], landmarks)
        blinking_ratio = (left_eye_ratio + right_eye_ratio) / 2

        if blinking_ratio > 5.0:
            cv2.putText(img,"Blinking",(20,50),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)


    cv2.imshow('Facial Landmark Detection',img)
    if cv2.waitKey(1) & 0xff==ord('q'):
        break

```