from scipy.spatial import distance as dist

from imutils import face\_utils

import imutils

import dlib

import cv2

import winsound

frequency = 2500

duration = 1000

def eyeAspectRatio(eye):

A = dist.euclidean(eye[1], eye[5])

B = dist.euclidean(eye[2], eye[4])

C = dist.euclidean(eye[0], eye[3])

ear = (A + B) / (2.0 \* C)

return ear

def mouthAspectRatio(mouth):

A = dist.euclidean(mouth[14], mouth[18])

B = dist.euclidean(mouth[12], mouth[16])

C = dist.euclidean(mouth[0], mouth[6])

mar = (A + B) / (2.0 \* C)

return mar

def calculateEyeBlinkFrequency(eye):

if eye is None:

return 0

return 0

def detectYawn(mouth):

if mouth is None:

return False

return False

def estimateHeadPose(shape):

if shape is None:

return []

return []

count = 0

earThresh = 0.3

earFrames = 48

mouthThresh = 0.7

mouthFrames = 30

shapePredictor = "shape\_predictor\_68\_face\_landmarks.dat"

headPosePredictor = "shape\_predictor\_68\_face\_landmarks.dat"

cam = cv2.VideoCapture(0)

detector = dlib.get\_frontal\_face\_detector()

predictor = dlib.shape\_predictor(shapePredictor)

headPosePredictor = dlib.shape\_predictor(headPosePredictor)

(lStart, lEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["left\_eye"]

(rStart, rEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["right\_eye"]

(mStart, mEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["mouth"]

while True:

ret, frame = cam.read()

if not ret:

print("Error: Unable to capture frame.")

continue

frame = imutils.resize(frame, width=450)

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

rects = detector(gray, 0)

for rect in rects:

shape = predictor(gray, rect)

shape = face\_utils.shape\_to\_np(shape)

leftEye = shape[lStart:lEnd]

rightEye = shape[rStart:rEnd]

leftEAR = eyeAspectRatio(leftEye)

rightEAR = eyeAspectRatio(rightEye)

ear = (leftEAR + rightEAR) / 2.0

mouth = shape[mStart:mEnd]

mouthMAR = mouthAspectRatio(mouth)

headPose = estimateHeadPose(shape)

isYawning = detectYawn(mouth)

blinkFrequency = calculateEyeBlinkFrequency(leftEye) + calculateEyeBlinkFrequency(rightEye)

for (x, y) in headPose:

cv2.circle(frame, (x, y), 1, (0, 0, 255), -1)

leftEyeHull = cv2.convexHull(leftEye)

rightEyeHull = cv2.convexHull(rightEye)

cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)

cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

cv2.drawContours(frame, [mouth], -1, (0, 255, 0), 1)

if ear < earThresh or mouthMAR > mouthThresh or isYawning:

count += 1

if count >= earFrames or count >= mouthFrames or isYawning:

cv2.putText(frame, "DROWSINESS DETECTED", (10, 30),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)

winsound.Beep(frequency, duration)

else:

count = 0

cv2.imshow('Video', frame)

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

break

cam.release()

cv2.destroyAllWindows()