|  |  |
| --- | --- |
| Team ID | NM2023TMID14700 |
| Project name | Drowsiness detection and alerting system |

from scipy.spatial import distance

from imutils import face\_utils

import imutils

import dlib

import cv2

import numpy as np

def eye\_aspect\_ratio(eye):

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

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

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

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

return ear

eye\_threshold = 0.25

consecutive\_frames = 20

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

predict = dlib.shape\_predictor("shape\_predictor\_68\_face\_landmarks.dat")

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

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

cap=cv2.VideoCapture(0)

count=0

while True:

ret, frame=cap.read()

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

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

subjects = detect(gray, 0)

for subject in subjects:

shape = predict(gray, subject)

shape = face\_utils.shape\_to\_np(shape)#converting to NumPy Array

leftEye = shape[lStart:lEnd]

rightEye = shape[rStart:rEnd]

left\_eye\_asp\_ratio = eye\_aspect\_ratio(leftEye)

right\_eye\_asp\_ratio = eye\_aspect\_ratio(rightEye)

eye\_asp\_ratio = (left\_eye\_asp\_ratio + right\_eye\_asp\_ratio) / 2.0

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)

if eye\_asp\_ratio < eye\_threshold:

count = count + 1

if count >= consecutive\_frames:

cv2.putText(frame, "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ALERT!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", (10, 30),

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

cv2.putText(frame, "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ALERT!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", (10,325),

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

else:

count = 0

cv2.imshow("Frame", frame)

k = cv2.waitKey(1) & 0xFF

if k == 27 or k==ord("q"):

break

cv2.destroyAllWindows()

cap.stop()