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

Count = 0

earThresh = 0.3

earFrames = 48

shapePredictor = “shape\_predictor\_68\_face\_landmarks.dat”

cam = cv2.VideoCapture(0)

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

predictor = dlib.shape\_predictor(shapePredictor)

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

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

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

leftEyeHull = cv2.convexHull(leftEye)

rightEyeHull = cv2.convexHull(rightEye)

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

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

if ear < earThresh:

count += 1

if count >= earFrames:

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()