

CODE

```
#Import necessary libraries
from scipy.spatial import distance
from imutils import face_utils
import numpy as np
import pygame #For playing sound
import time
import dlib
import cv2

#Initialize Pygame and load music
pygame.mixer.init()
pygame.mixer.music.load('C:\\Users\\VIKAS\\OneDrive\\Desktop\\Driver-Drowsiness-Detector-master\\Driver-Drowsiness-Detector-master\\audio\\alert.wav')
#Minimum threshold of eye aspect ratio below which alarm is triggered
EYE_ASPECT_RATIO_THRESHOLD = 0.3
#Minimum consecutive frames for which eye ratio is below threshold for alarm to be triggered
EYE_ASPECT_RATIO_CONSEC_FRAMES = 50
#Counts no. of consecutive frames below threshold value
COUNTER = 0
#Load face cascade which will be used to draw a rectangle around detected faces.
face_cascade = cv2.CascadeClassifier('C:\\Users\\VIKAS\\OneDrive\\Desktop\\Driver-Drowsiness-Detector-master\\Driver-Drowsiness-Detector-master\\haarcascade_frontalface_default.xml')
#This function calculates and return eye aspect ratio
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*C)
    return ear
#Load face detector and predictor, uses dlib shape predictor file
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor('C:\\Users\\VIKAS\\OneDrive\\Desktop\\Driver-Drowsiness-Detector-master\\Driver-Drowsiness-Detector-master\\shape_predictor_68_face_landmarks\\shape_predictor_68_face_landmarks.dat')

#Extract indexes of facial landmarks for the left and right eye
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS['left_eye']
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS['right_eye']
#Start webcam video capture
video_capture = cv2.VideoCapture(0)
#Give some time for camera to initialize(not required)
time.sleep(1)
while(True):
    #Read each frame and flip it, and convert to grayscale
    ret, frame = video_capture.read()
    frame = cv2.flip(frame,1)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```

#Detect facial points through
detector functionfaces =
detector(gray, 0)
#Detect faces through
haarcascade_frontalface_default.xml
face_rectangle =
face_cascade.detectMultiScale(gray, 1.3, 5)
#Draw rectangle around each face detected
for (x,y,w,h) in face_rectangle:
    cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0)
    ,2)
#Detect
facial
points
for face
in faces:
    shape = predictor(gray, face)
    shape = face_utils.shape_to_np(shape)

    #Get array of coordinates of leftEye
    and rightEyeleftEye =
    shape[lStart:lEnd]
    rightEye = shape[rStart:rEnd]

    #Calculate aspect ratio of both eyes
    leftEyeAspectRatio =
    eye_aspect_ratio(leftEye)
    rightEyeAspectRatio =
    eye_aspect_ratio(rightEye)

    eyeAspectRatio = (leftEyeAspectRatio + rightEyeAspectRatio) / 2

    #Use hull to remove convex contour discrepancies and draw eye shape
    around eyesleftEyeHull = 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)

    #Detect if eye aspect ratio is less than threshold
    if(eyeAspectRatio <
    EYE_ASPECT_RATIO_THRESHOLD):
        COUNTER += 1
        #If no. of frames is greater than threshold frames,
        if COUNTER >= EYE_ASPECT_RATIO_CONSEC_FRAMES:
            pygame.mixer.music.play(-1)
            cv2.putText(frame, "You are Drowsy", (150,200), cv2.FONT_HERSHEY_SIMPLEX, 1.5,

```

```
(0,0,255), 2)else:
    pygame.mixer.music.stop()
    COUNTER = 0
#Show video
feed
cv2.imshow('V
ideo', frame)
if(cv2.waitKey(1) & 0xFF
    == ord('q')):break

#Finally when video capture is over, release the video capture and
destroyAllWindowsvideo_capture.release()
cv2.destroyAllWindows()
```