**REAL-TIME DRIVER DROWSINESS DETECTION THROUGH DEEP LEARNING**

**Source Code:**

import cv2

import numpy as np

import dlib

from imutils import face\_utils

# Initialize the camera

cap = cv2.VideoCapture(0)

# Initialize face detector and facial landmark predictor

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

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

# Status flags

sleep = 0

drowsy = 0

active = 0

status = ""

color = (0, 255, 0)  # Green for "awake"

def compute(ptA, ptB):

    return np.linalg.norm(ptA - ptB)

def eye\_aspect\_ratio(eye\_points):

    # Calculate Eye Aspect Ratio (EAR)

    A = compute(eye\_points[1], eye\_points[5])  # Vertical distance

    B = compute(eye\_points[2], eye\_points[4])  # Vertical distance

    C = compute(eye\_points[0], eye\_points[3])  # Horizontal distance

    ratio = (A + B) / (2.0 \* C)

    return ratio

while True:

    \_, frame = cap.read()

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

    faces = detector(gray)

    for face in faces:

        landmarks = predictor(gray, face)

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

        # Extract eye landmarks

        left\_eye = landmarks[36:42]

        right\_eye = landmarks[42:48]

        # Compute EAR for both eyes

        left\_ear = eye\_aspect\_ratio(left\_eye)

        right\_ear = eye\_aspect\_ratio(right\_eye)

        ear = (left\_ear + right\_ear) / 2.0

        # Check if eyes are closed based on EAR thresholds

        if ear < 0.20:  # Drowsy threshold

            sleep += 1

            drowsy = 0

            active = 0

            if sleep > 6:

                status = "DROWSINESS ALERT!"

                color = (0, 0, 255)  # Red

        elif 0.20 <= ear <= 0.25:  # Intermediate threshold

            drowsy += 1

            sleep = 0

            active = 0

            if drowsy > 6:

                status = "Drowsy"

                color = (0, 255, 255)  # Yellow

        else:  # Awake threshold

            active += 1

            sleep = 0

            drowsy = 0

            if active > 6:

                status = "THE DRIVER IS AWAKE"

                color = (0, 255, 0)  # Green

        # Display status on frame

        cv2.putText(frame, status, (10, 30), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, color, 2)

        cv2.putText(frame, f"Eye Aspect Ratio: {ear:.2f}", (10, 60), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 255, 0), 2)

        # Draw eye landmarks

        for (x, y) in np.concatenate([left\_eye, right\_eye]):

            cv2.circle(frame, (x, y), 1, (0, 255, 0), -1)  # Green dots on eyes

    cv2.imshow("Frame", frame)

    # Break on pressing ESC key

    key = cv2.waitKey(1)

    if key == 27:

        break

cap.release()

cv2.destroyAllWindows()