Drowsy Driver Detection System Using Deep Learning

Code:

```
import cv2
import dlib
from scipy.spatial import distance
from pygame import mixer
from imutils import face_utils
# Initialize Pygame Mixer for audio alerts
mixer.init()
mixer.music.load(r"C:\Users\Admin\Desktop\driver\sunrise-and-suncastles-321413.mp3")
# Constants
EYE_AR_THRESH = 0.22
EYE_AR_CONSEC_FRAMES = 3
MOUTH_AR_THRESH = 0.7
# Landmark indices
(lStart, lEnd) = (42, 48) # Left eye
(rStart, rEnd) = (36, 42) # Right eye
(mStart, mEnd) = (60, 68) # Inner mouth
# Functions
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])
return (A + B) / (2.0 * C)
def mouth_aspect_ratio(mouth):
```

```
A = distance.euclidean(mouth[1], mouth[7])
B = distance.euclidean(mouth[2], mouth[6])
C = distance.euclidean(mouth[0], mouth[4])
return (A + B) / (2.0 * C)
# Flags
eye_counter = 0
song_playing = False
# Dlib face detector and predictor
detect = dlib.get_frontal_face_detector()
predict =
dlib.shape_predictor(r"C:\Users\Admin\Desktop\driver\shape_predictor_68_face_landmarks.dat")
# Start webcam
cap = cv2.VideoCapture(0)
while True:
ret, image = cap.read()
if not ret:
break
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces = detect(gray, 0)
for face in faces:
shape = predict(gray, face)
shape = face_utils.shape_to_np(shape)
```

```
left_eye = shape[lStart:lEnd]
right_eye = shape[rStart:rEnd]
mouth = shape[mStart:mEnd]
# EAR calculation
left_ear = eye_aspect_ratio(left_eye)
right_ear = eye_aspect_ratio(right_eye)
ear = (left_ear + right_ear) / 2.0
# MAR calculation
mar = mouth_aspect_ratio(mouth)
# Drowsiness detection
if ear < EYE_AR_THRESH:
eye_counter += 1
if eye_counter >= EYE_AR_CONSEC_FRAMES and not song_playing:
mixer.music.play()
song_playing = True
elif mar > MOUTH_AR_THRESH:
if not song_playing:
mixer.music.play()
song_playing = True
else:
eye_counter = 0
if song_playing:
mixer.music.stop()
song_playing = False
```

```
# Draw alert text if drowsy
if song_playing:
cv2.putText(image, "DROWSINESS DETECTED!", (120, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 0, 255), 3)
# Draw contours
cv2.drawContours(image, [cv2.convexHull(left_eye)], -1, (0, 255, 0), 1)
cv2.drawContours(image, [cv2.convexHull(right_eye)], -1, (0, 255, 0), 1)
cv2.drawContours(image, [cv2.convexHull(mouth)], -1, (255, 0, 0), 1)
# Display EAR and MAR on screen
cv2.putText(image, f"EAR: {ear:.2f}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 255, 255), 2)
cv2.putText(image, f"MAR: {mar:.2f}", (10, 60),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 255, 255), 2)
# Show frame
cv2.imshow("Frame", image)
if cv2.waitKey(1) & 0xFF == ord('q'):
break
# Cleanup
cap.release()
cv2.destroyAllWindows()
```

Screenshots:

Figure A: EAR and YAWN

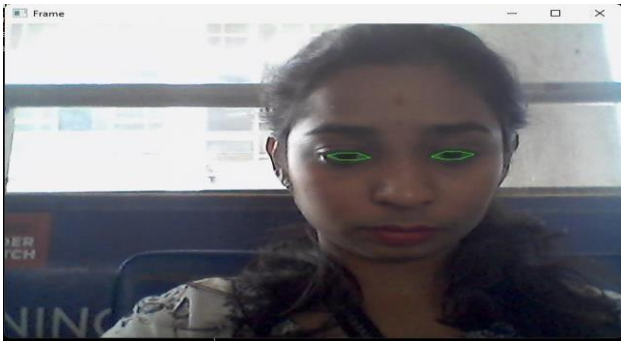

Figure B: EYE Aspect

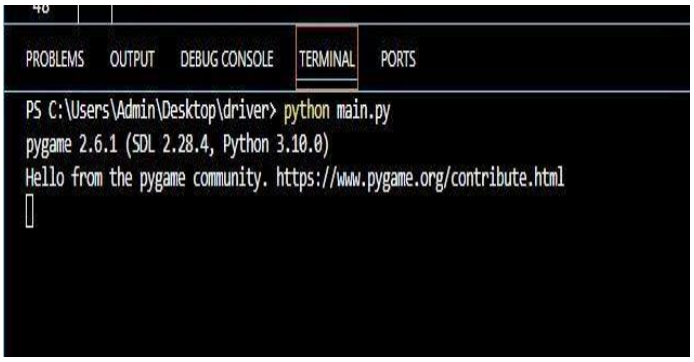

Figure B: Loading Facial Landmarking Prediction