Driver Drowsiness Detection System

Computer Aided Vehicle
Design and Safety

Ayush Tankha
2k17/AE/13

Tools Used

Python Libraries -

- OpenCV
- Numpy
- Dlib
- Imutils

Laptop Camera

Code

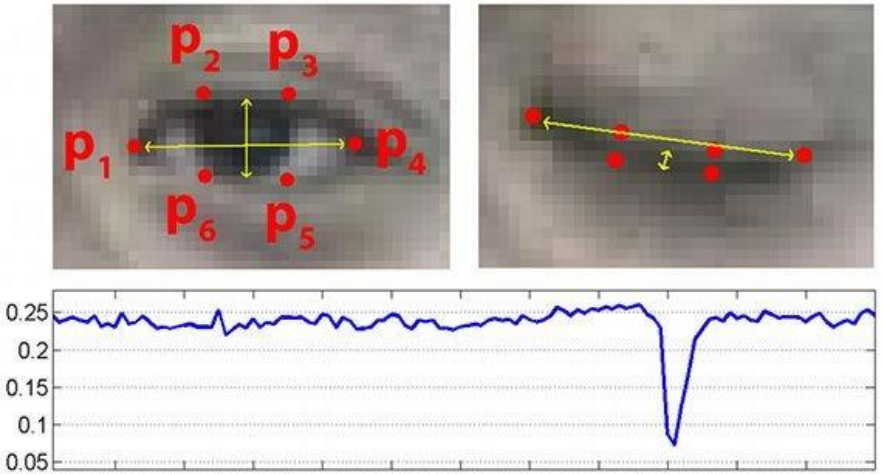
```
from scipy.spatial import distance
from imutils import face_utils
import imutils
import dlib
import cv2

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

thresh = 0.25
frame_check = 20
detect = dlib.get_frontal_face_detector()
predict = dlib.shape_predictor(".\shape_predictor_68_face_landmarks.dat")# Dat file is the crux of the code

(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["right_eye"]
cap=cv2.VideoCapture(0)
flag=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]
			leftEAR = eye_aspect_ratio(leftEye)
			rightEAR = eye_aspect_ratio(rightEye)
			ear = (leftEAR + rightEAR) / 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 ear < thresh:
				flag += 1
				print (flag)
				if flag >= frame_check:
					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)
					#print ("Drowsy")
			else:
				flag = 0
		cv2.imshow("Frame", frame)
		key = cv2.waitKey(1) & 0xFF
		if key == ord("q"):
			break
cv2.destroyAllWindows()
cap.stop()
```

Working

SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python.
Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV.
dlib is a toolkit for making real world machine learning and data analysis applications in C++. While the library is originally written in C++, it has good, easy to use Python bindings.
OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

First the program uses a machine learning face detection model to detect landmarks on the driver's face. Each eye is represented by 6 (x,y) coordinates.

It uses the landmarks around the eyes to calculate the eye aspect ratio. When the distance remains below a certain threshold value for 20 frames, the program issues an alert.



Algorithm Improvements

- We can more features like tilt of head, yawn frequency, blink frequency, positioning of hands, how often someone rubs their eyes, etc

- Real life implementation will probably be done using a conjunction of wide variety of algorithms. It is highly likely that the most important algorithms will be deep learning algorithms trained on huge datasets.

Hardware Improvements for implementation

- One could use an array of different types cameras suitable for dynamic lighting situations in real world situations. Eg- Thermal, LIDAR, IR, etc.

- Tactile and sound sensors for detecting behaviour patterns like slurring, slouching, etc with additional data and hence greater accuracy.

- Integration with car processor.

AI and Trust

We already have technology stack advanced enough to implement this on a large scale but there are some challenges that have to be solved first. First is cost. Installation of high definition cameras and sensors with a processing unit will drive the cost of the vehicle up. We are also facing a silicon shortage at the moment. Secondly, there needs to be active conversation between consumers, government and manufacturers about data collection & storage as well as privacy. Our automobiles are immensely personal spaces that not everyone may wish to share. Transparency about the system and the data may help the case but there will always certainly be a trade off between privacy and AI. At the end of the day, individual rights imply that control should be in the hands of the consumer but their limited understanding of this technology means they have to rely on trust.

Thank You

Submitted to:

Prof Navriti Gupta