DRIVER'S DROWSINESS DETECTION SYSTEM

Submitted by

RUPESH (2106005)

TILAK KUMAR (2106032)
PIYUSH KUMAR DAS
(2106031)

NISHANT KUMAR (2106028)

CONTENTS

Overview

Driver drowsiness detection is a car safety technology which helps prevent accidents caused by the driver getting drowsy. Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads. The drowsiness detection system is capable of detecting drowsiness quickly. The driver behaviors are noticed in many conditions such as wearing spectacles and also in the dark condition

inside the vehicle. The system is capable of detecting the drowsiness condition within the duration of more than two seconds. After the detection of abnormal behaviors, it is alerted to the driver through alarms and the parking lights will be on that will stop the vehicle which reduces the accidents due to drowsiness of the driver. A deep learning Architecture detects the face and eyes, based on the status of the eyes. If the eyes are closed more than usual time, it generates an alarm, intimating the driver. Neglecting our duties towards safer travel has enabled hundreds of thousands of tragedies to get associated with this wonderful invention every year. In order to monitor and prevent a destructive outcome from such negligence, many researchers have written research papers on driver drowsiness detection systems. But at times, some of the points and observations made by the system are not accurate enough .

Motivation

- Witnessing accidents.

- Bringing awareness.
- Taking advantage of Technology.
- Bringing up Suitable Software.
- Include the Software into convenience.
- Looking for updating the versions

.

Goal and Objectives

There are many products out there that provide the measure of fatigue level in the drivers which are implemented in many vehicles. Also, it alerts the user on reaching a certain saturation point of the drowsiness measure.

SCOPE

This project can be integrated with car, so that automatic speed control can be imparted if the driver is found sleeping.

PREREQUISITES

The requirement for this Python project is a webcam or computer camerathrough which we will capture images. You need to have Python (3.6 version recommended) installed on your system, then using pip, you can install the necessary package.

PROJECT DESCRIPTION

• EXISTING SYSTEM AND DISADVANTAGES

- **Vision based techniques:**

NO EYE DETECTION – most critical sign of drowsiness Yawning and nodding are not always practical. Varies from person to person – some may not yawn when they are sleepy sometimes.

- **Physiological sensors**:

More accurate solutions

Needs to be attached to the human body

- If driver forgets to wear it?
- May hesitate to wear

PROPOSED SYSTEM AND ADVANTAGES

The proposed system will be continuously monitoring the movement of the driver's eye by a live camera and all the monitored signals are pre - processed.

In order to overcome drawbacks, Python is used in which the trained system is already installed and avoids the time to process that occurs from the scratch.

A Black Box with the software installed is used to detect the driver drowsiness and alerts the driver with buzzer, if driver is affected by drowsiness.

PROJECT REQUIREMENTS

Software Requirements

Python: Python is the basis of the program that we wrote. It utilizes many of the python libraries.

Libraries:

- cv2
- numpy
- dlib
- imutils
- serial
- time

Operating System:

- Windows

○ Hardware Requirements

- Laptop with basic hardware
- Arduino UNO
- LED
- 16*2 LCD
- Buzzer

IMPLEMENTATION

1. Take image as input from a camera.
2. Detect the face in the image and create a Region of Interest(ROI).
3. Detect the eyes from ROI and feed it to the classifier.
4. Classifier will categorize whether eyes are open or closed.
5. Calculate score to check whether the person is drowsy.

First we have used a camera which is setup at desirable position in a car that looks for faces stream.

If face gets detected, the facial landmark detection task is applied and region of eyes is extracted.

Once we get the eye region, we calculate the Eye Aspect Ratio to find out if the eyelids are down for a substantial amount of time.

On the off chance that the Eye Aspect Ratio demonstrates that the eyes are shut for a considerably long measure of time, the alert will sound noisy to wake the driver up.For the functionalities of the system and to make it work efficiently we have used OpenCv, dlib and Python. The implementation of the drowsiness detector system includes machine learning algorithms which are in turn included in OpenCv ML algorithms. There are numerous ML algorithms but for our purpose we required only the face detector algorithm. It works efficiently well overall. It can also be used to detect various different types of objects with the required software.

CODE

1. Arduino code

```
#include <LiquidCrystal.h>



LiquidCrystal lcd(2, 3, 4, 5, 6,7);
const int buzzer_Pin = 8;
const int led_Pin = 9;
char sleep_status = 0;

void setup() {
  Serial.begin(9600);
  pinMode(buzzer_Pin, OUTPUT);
  pinMode(led_Pin, OUTPUT);
  lcd.begin(16, 2);
  lcd.print("Driver Sleep ");
  lcd.setCursor(0,2);
  lcd.print("Detection SYSTEM");
  digitalWrite(buzzer_Pin, LOW);
  digitalWrite(led_Pin, LOW);
}
```

```
void loop()
{
    while (Serial.available() > 0)
  {
    sleep_status = Serial.read();
    if(sleep_status == 'a')
    {
        lcd.clear();
        lcd.print("Please wake up");
        digitalWrite(buzzer_Pin, HIGH);
        digitalWrite(led_Pin, HIGH);
        delay(2000);
        digitalWrite(buzzer_Pin, LOW);
        digitalWrite(led_Pin, LOW);

        delay(100);
    }
    else if(sleep_status == 'b')
    {
        lcd.clear();
        lcd.print("All Ok");
        lcd.setCursor(0,2);
        lcd.print("Drive Safe");
        digitalWrite(buzzer_Pin, LOW);
        digitalWrite(led_Pin, LOW);
        delay(2000);
    }
    else
    {
      /* Do Nothing */
    }
  }
}
```

2. Python code

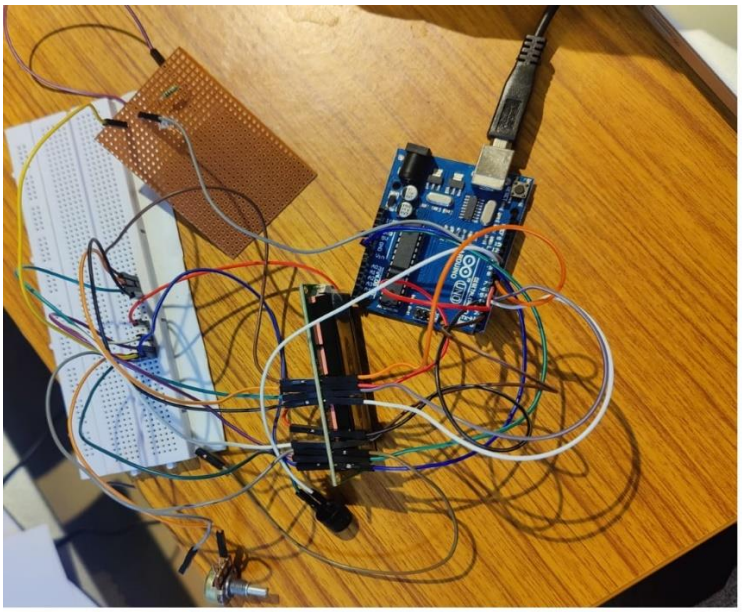
```
import cv2

import numpy as np
import dlib
from imutils import face_utils
import serial
import time

s = serial.Serial('COM7', 9600)
```

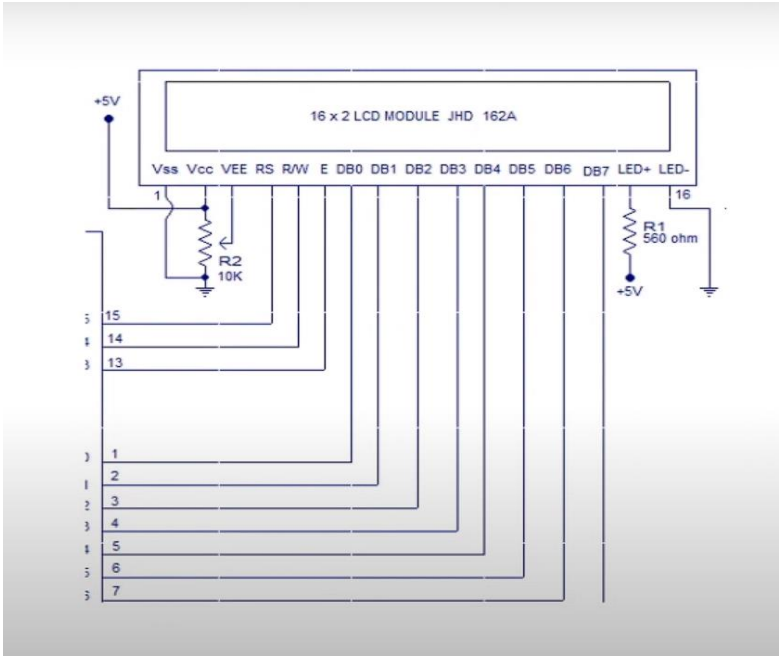
```
cap = cv2.VideoCapture(0)

hog_face_detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

sleep = 0
drowsy = 0
active = 0
status = ""
color = (0, 0, 0)

def compute(ptA, ptB):
    dist = np.linalg.norm(ptA - ptB)
    return dist

def blinked(a, b, c, d, e, f):
    up = compute(b, d) + compute(c, e)
    down = compute(a, f)
    ratio = up / (2.0 * down)

    if ratio > 0.25:
        return 2
    elif 0.21 < ratio <= 0.25:
        return 1
    else:
        return 0

while True:
    ret, frame = cap.read()
    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = hog_face_detector(gray)

    for face in faces:
        x1 = face.left()
        y1 = face.top()
        x2 = face.right()
        y2 = face.bottom()

        face_frame = frame.copy()
        cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

        landmarks = predictor(gray, face)
        landmarks = face_utils.shape_to_np(landmarks)

        left_blink = blinked(landmarks[36], landmarks[37],
                             landmarks[38], landmarks[41],
```

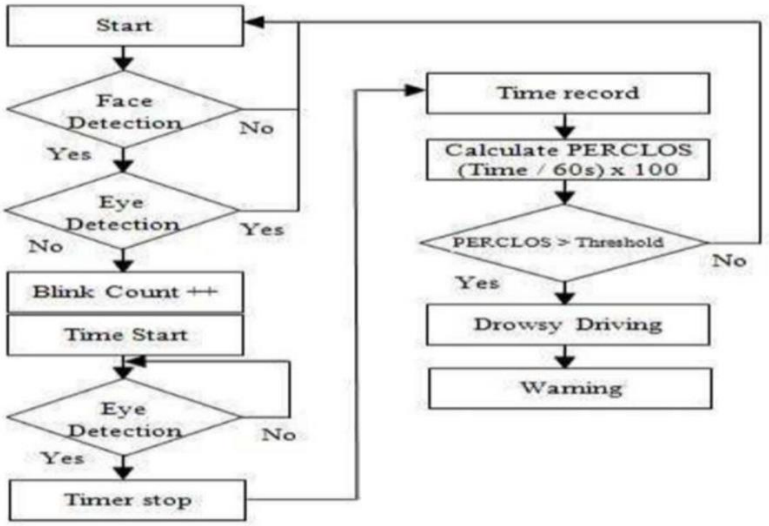
```
                        landmarks[40], landmarks[39])

    right_blink = blinked(landmarks[42], landmarks[43],
                          landmarks[44], landmarks[47],
                          landmarks[46], landmarks[45])

    if left_blink == 0 or right_blink == 0:
        sleep += 1
        drowsy = 0
        active = 0
        if sleep > 6:
            s.write(b'a')
            time.sleep(2)
            status = "SLEEPING !!!"
            color = (0, 0, 255)

    elif left_blink == 1 or right_blink == 1:
        sleep = 0
        active = 0
        drowsy += 1
        if drowsy > 6:
            s.write(b'a')
            time.sleep(2)
            status = "Drowsy !"
            color = (0, 0, 255)

    else:
        drowsy = 0
        sleep = 0
        active += 1
        if active > 6:
            s.write(b'b')
            time.sleep(2)
            status = "Active :)"
            color = (0, 0, 255)

    cv2.putText(frame, status, (100, 100), cv2.FONT_HERSHEY_SIMPLEX, 1.2, color, 3)

    for (x, y) in landmarks:
        cv2.circle(face_frame, (x, y), 1, (255, 255, 255), -1)

cv2.imshow("Frame", frame)
# Uncomment the line below to visualize facial landmarks
# cv2.imshow("Result of detector", face_frame)

key = cv2.waitKey(1)
if key == 27:
    break
```

```
cap.release()
cv2.destroyAllWindows()
```

Circuit Design







CONCLUSION AND FUTURE WORK

CONCLUSION:

It completely meets the objectives and requirements of the system. The framework has achieved an unfaltering state where all the bugs have been disposed of. The framework cognizant clients who are familiar with the framework and comprehend it's focal points and the fact that it takes care of the issue of stressing out for individuals having fatiguerelated issues to inform them about the drowsiness level while driving.

The ultimate goal of the system is to check the drowsiness condition of the driver. Based on the eye movements of the driver, the drowsiness is detected and according to eye blink, the alarm will be generated to alert the driver and to reduce the speed of the vehicle along with the indication of parking light. By doing this, many accidents will be reduced and

provides safety to the driver and vehicle. A system that is driver safety and car security is presented only in luxurious costly cars. Using eye detection, driver security and safety can be implemented in normal car also.

FUTURE WORK:

The model can be improved incrementally by using other parameters like blink rate, yawning, state of the car, etc. If all these parameters are used it can improve the accuracy by a lot.

We plan to further work on the project by adding a sensor to track the heart rate in order to prevent accidents caused due to sudden heart attacks to drivers.

Same model and techniques can be used for various other uses like Netflix and other streaming services can detect when the user is asleep and stop the video accordingly. It can also be used in application that prevents user from sleeping .