Main Project Report

on

DROWSINESS DETECTION USING OPENCV

submitted by

Anakha K Babu (12160012)
Bhagya C (12160027)
Jithinya K (12160043)

In partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering.

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

MARCH 2019

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

Certified that this is a bonafide record of the Major Project titled

DROWSINESS DETECTION USING OPENCV

done by
Anakha K Babu (12160012)
Bhagya C (12160027)
Jithinya K (12160043)

of VIII Semester, Computer Science and Engineering in the year 2019 in partial fulfillment requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering of Cochin University of Science and Technology.

Dr.Latha R Nair	Dr.M.Sudheep Elayidom	Ms.Lino Murali
Head of Division	Project Coordinator	Project Guide

Acknowledgement

We take this opportunity to thank the supreme being, the source of all knowledge whose blessings are our guiding light in any venture we take up. Were in short of words to express our gratitude to Ms.Lino Murali, our project guide who guided us and helped us constantly with her inputs and suggestion without which we couldnt have implemented this project the way it is working today. We are highly indebted to Dr.M. Sudheep Elayidom , for his constant supervision and support in completing this project. We also express our heartfelt thanks to all other faculties and staffs advicers for helping us by providing all the necessary amenities for completing the project. A bouquet of gratitude to Dr.Latha R Nair, head of the division of Computer Science and Engineering for all kinds of encouragement extended to us.

Anakha K Babu(12160012)
Bhagya C(12160027)
Jithinya K(12160043)

Declaration

We, Ms.Anakha K Babu , Ms. Bhagya C,& Ms. Jithnya K hereby declare that this project is the record of authentic work carried out by us during the academic year 2018 - 2019 and has not been submitted to any other University or Institute towards the award of any degree.

Abstract

Driver fatigue is one of the major causes of accidents in the world. Detecting the drowsiness of the driver is one of the surest ways of measuring driver fatigue. In this project we aim to develop a prototype drowsiness detection system. This system works by monitoring the eyes of the driver by a camera and sounding an alarm when he/she is drowsy.The system so designed is a non-intrusive real-time monitoring system. The priority is on improving the safety of the driver without being obtrusive. In this project the eye blink of the driver is detected. If the drivers eyes remain closed for more than a certain period of time, the driver is said to be drowsy and an alarm is sounded. The programming for this is done in OpenCV using the facial landmark localization for the detection of facial features.

Contents

List of Figures

Chapter 1

Introduction

Generally, there are many reasons behind highway traffic accidents. Driver drowsiness is one of the major causes of serious traffic accidents. According to the National Highway Traffic Safety Administration (NHTSA) [1], there are about 56,000 crashes caused by drowsy drivers every year in US, which results in about 1,550 fatalities and 40,000 nonfatal injuries annually. The National Sleep Foundation also reported that 60The aim of this project is to develop a prototype drowsiness detection system. The focus will be placed on designing a system that will accurately monitor the open or closed state of the drivers eyes in real-time. By monitoring the eyes, it is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident.

Chapter 2

Literature Review

2.1 Real-Time Eye Blink Detection using Facial Landmarks

Real-Time Eye Blink Detection using Facial Landmarks by Tereza Soukupova and Jan Cech introduced A real-time algorithm to detect eye blinks in a video sequence from a standard camera is proposed. Recent landmark detectors, trained on in-the wild datasets exhibit excellent robustness against a head orientation with respect to a camera, varying illumination and facial expressions. We show that the landmarks are detected precisely enough to reliably estimate the level of the eye opening. The proposed algorithm therefore estimates the landmark positions, extracts a single scalar quantity eye aspect ratio (EAR) characterizing the eye opening in each frame. Finally, an SVM classifier detects eye blinks as a pattern of EAR values in a short temporal window. The simple algorithm outperforms the state-of-the-art results on two standard dataset.State-of-the-art on two standard datasets was achieved using the robust landmark detector followed by a simple eye blink detection based on the SVM. The algorithm runs in real-time, since the additional computational costs for the eye blink detection are negligible besides the real-time landmark detectors. It finalised as the proposed SVM method that uses a temporal window of the eye aspect ratio (EAR), outperforms the EAR thresholding. On the other hand, the thresholding is usable as a single image classifier to detect the eye state, in case that a longer sequence is not available. It points out a limitation that a fixed blink duration for all subjects was assumed, although everyones blink lasts differently. The results could be improved by an adaptive approach. Another limitation is in the eye

opening estimate. While EAR is estimated from a 2D image, it is fairly insensitive to a head orientation, but may lose discriminability for out of plane rotations. A solution might be to define the EAR in 3D. There are landmark detectors that estimate a 3D pose (position and orientation) of a 3D model of landmark.

2.2 Drowsy Driver Detection System Using Eye Blink Patterns

An automatic drowsy driver monitoring and accident prevention system that is based on monitoring the changes in the eye blink duration. The proposed method detects visual changes in eye locations using the proposed horizontal symmetry feature of the eyes. The new method detects eye blinks via a standard webcam in real-time at 110fps for a 320240 resolution. Experimental results in the JZU [3] eye-blink database showed that the proposed system detects eye blinks with a 94The new method to detect eye blinks uses the symmetry property. The proposed system is independent of the head movements as it works within the same frame. Therefore, it has an advantage over the other systems that use statistical information from the past frames. In addition, it runs at a 110fps rate on Intel Xeon 2.9 GHz CPU for a 320240 resolution which is acceptable for real-time scenarios and leaves time for other tasks. Unfortunately, no common database exists for comparing our results for drowsiness; therefore only the results for eye-blink detection are given. The proposed system detects eye blinks with a 94

Chapter 3

System Analysis

3.1 Existing System

In-vehicle camera is commonly installed to realize the possible reasons of car accidents. Such a camera can also be used to detect the fatigue of the driver. Several studies related to the fatigue detection are described as follows. Sharma et a utilized the number of pixels in the eye image to determine the eye state, open or close. Hornget at. established an edge map to locate the eyes locations and the eye state is determined based on the HSL color space of the eye image. Its accuracy is dependent on the location of the eyes. Sharma and Banga converted the face image to YCbCr color space. The average and standard deviation of the pixel number in the binarization image is computed. Then, fuzzy rules [5] are used to determine the eye state. Liu et al. and Tabrizi et al. proposed methods to detect the upper and lower eyelids based on the edge map. The distance between the upper and lower eyelids is then used to analyze the eye state. Besides, Dong et al. and Li et al. proposed methods by utilizing AAM (Active Appearance Model) to locate the eyes. Then, a PERCLO (PERcentage of eye CLOsure) was computed to detect the fatigue. For the above methods, the locating of eye areas was easily influenced by the change of brightness. Circular Hough transform is popular method to overcome the influence of brightness. Several studies proposed methods to locate the pupil of eyes by using circular Hough transform. Then,the eye state was analyzed according to the locations of pupils. Zhengpei calculated the ratio of eye closing during a period of time. The ratio can reflect drivers vigilance level . Wenhui Dong proposed a method to detect the distance of

eyelid, and then judged the drivers status by this kind of information [14]. Nikolaos P used front view and side view images to precisely locate eyes . Edge detection and gray-level projection methods were also applied for the eyes location by Wen-Bing Horng. Zutao Zhang located the face by using Haar algorithm and proposed an eye tracking method based on Unscented Kalman Filter . Abdel Fattah Fawky presented a combination of algorithms, namely wavelets transform, edge detection and YCrCb transform in the eye detection . QiangJi depended on IR illumination to locate eyes . Eyes always contain two kinds of information: size of opening and duration of the different states. By analyzing the change rules of eyes in fatigue, by extracting eyes using open cv methods we propose an efficient approach for driver fatigue detection.

3.2 Proposed System

In proposed method, first the image is acquired by the webcam for processing. The images of the driver are captured from the camera which is installed in front of the driver on the car dashboard. It will be passed to preprocessing which prepares the image for further processing by the system. Its main operations are to eliminate noises caused by the image acquisition subsystem and image enhancement using Histogram Equalization. Then we search and detect the faces in each individual frame. If no face is detected then another frame is acquired. If a face is detected, then a region of interest in marked within the face. This region of interest contains the eyes. Defining a region of interest significantly reduces the computational requirements of the system. After that the eyes are detected from the region of interest. If an eye is detected then there is no blink and the blink counter is set to 0. If the eyes are closed in a particular frame, then the blink counter is Incremented and a blink is detected. When the eyes are closed for more than 4 frames then it is deducible that the driver is feeling drowsy. Hence drowsiness is detected and an alarm sounded. After that the whole process is repeated as long as the driver is driving the car. The overall flowchart for drowsiness detection system is shown in Figure 1.

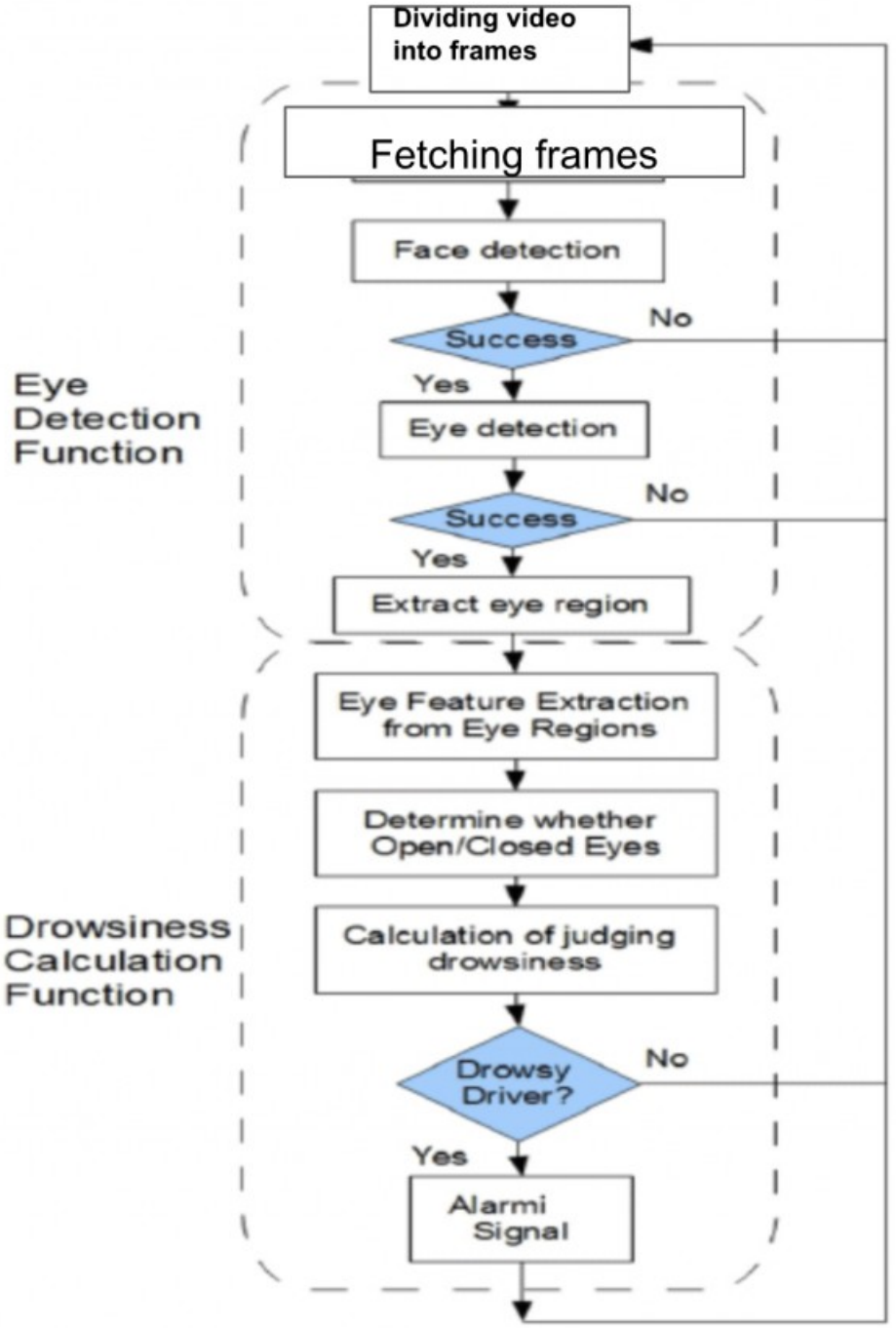

Figure 3.1: Flowchart

Chapter 4

System Study

In this section, we are going to present the SRS, system objectives and hardware and software tool requirements.

4.1 Software Requirements Specification

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform.

Purpose

The purpose of this project is to detect the drowsiness. . By monitoring the eyes, it is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident.We are dealing with real time situation where video is recorded and has to be processed. A specified algorithm is for detection of face in every frame. By face detection we means that locating the face in a frame or in other words finding location of facial characters through a HOG+ linear SVM object detector. The face region needs to be detected for further processing using facial landmark localization algorithm.Performs the detection of eye using landmark algorithm. Calculate the Eye Aspect Ratio (EAR) value to know the status of the eye. When EAR is matched with the reference or threshold value for deciding the state of the driver. If the threshold value is reducing,it means the diver feels sleepy. So sound an alarm.

Project Overview

The project has following functionalities:

- Facial landmark detection on images: dlibs HOG+Linear SVM object detector for face detection and then load the facial landmarks.
- Facial landmark detection on input video stream : Detecting facial landmarks in a series of frames by using dlibs HOG+Linear SVM object detector
- Extraction of eye regions : Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye , and then working clockwise around the remainder of the region.
- Computing the eye aspect ratio(EAR): EAR computation is done by using euclidean distance between the vertical eye landmark divided by euclidean distance between horizontal eye landmark.
- Counting the number of blinks: The eye aspect ratio falls below a certain threshold and then rises above the threshold, then it is register as blink. By default take the value 0.3.
- Sound an alarm when drowsiness detected: If the total number of consecutive frames is exceeds a threshold value then we assume the person is starting to doze off. Then playing the alarm sound to alert the driver.

Functional Requirements

- Facial landmark detection on images
  Purpose: To detect the facial landmark on images
  Actor: System
  Input: Image
  Output: Facial landmarks of the images appears.
- Facial landmark detection on input video stream
  Purpose: To detect the facial landmark on input video stream
  Actor:System
  Input: Input video

Output: Facial landmarks of the video stream

- Extraction of eye regions
  Purpose: To extract the eye region from the facial landmarks Actor: System
  Input: Live video
  Output: Extracted view of eye region obtained

- Computing the eye aspect ratio(EAR)

  Purpose: To compute the eye aspect ratio there by closing and opening of eye obtained
  Actor: System
  Input: Live video
  Output: Shows the EAR

- Counting the number of blinks
  Purpose: To count the number of blinks
  Actor: System
  Input: Input video
  Output: Shows the number of blinks

- Sound an alarm when drowsiness detected
  Purpose: When the drowsiness is detected then sound an alarm
  Actor: System
  Input: Live video
  Output: Alarm is sounded when drowsiness is detected

Non Functional Requirements

1. Performance Requirements

The drowsiness is detected while the driver is wearing spectacle. And it is working under the dim light,also during the night.

2. Safety Requirements

   The system must not be damaged or manipulated .

4.2 Hardware and Software Requirements

Hardware Requirements

- Laptop-Mac book pro
- Camera-Logitech C920: It is relatively affordable. Can shoot in full 1080p and it is plug-and-play compatible with nearly every device.

Software Requirements

- Python3 Interpreter
- OpenCV and Dlib libraries

Chapter 5

System Design

5.1 Introduction

Designing requires a careful planning and thinking on the part of the system designer. Designing a system means to plan how the various parts of it are going to achieve the desired goal. After the software requirements have been analysed and specified, design is the first of the three technical activities. Designing, coding and testing are required to build and verify the website.

5.2 Data Flow Diagrams

Data Flow Diagram is a pictorial way of showing the flow of data into/within the system, around the system and out of the system. It is a graphical representation of flow of data within a system. Unlike flowcharts, DFDs do not give detailed descriptions of modules but graphically describe data and how the data interact with the system. The DFD enable us to visualize how the system operates, its final output and the implementation of the system as a whole including modification if any. The purpose of data flow diagram is to provide a semantic bridge between users and system developers.

Level 0

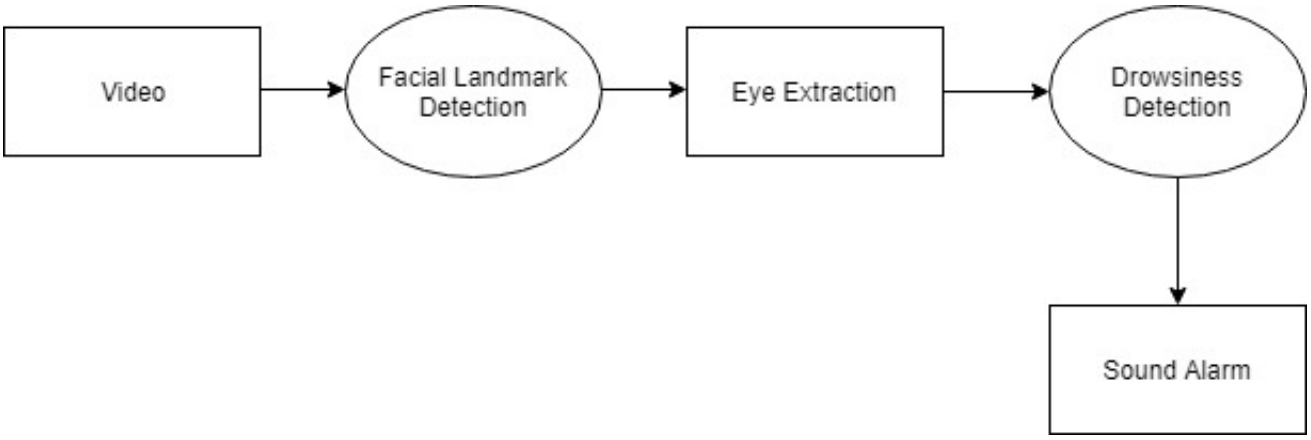

Figure 5.1: Level 0

Level 1 -Facial landmark detection

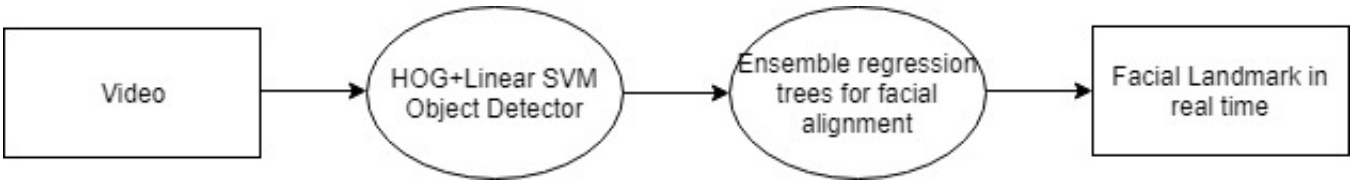

Figure 5.2: Level 1

Level 1-Drowsiness detection

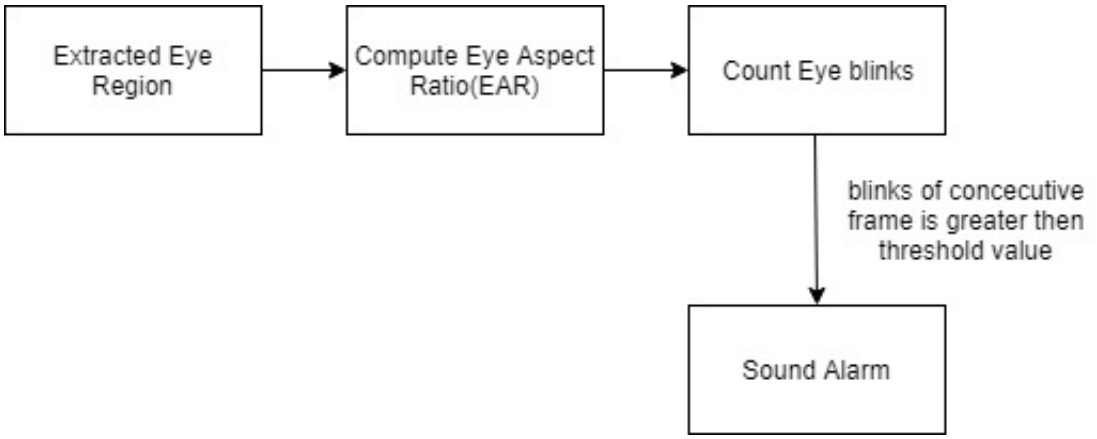

Figure 5.3: Level 1

5.3 Modular Design

The project is divided into different modules for drowsiness detection.

- Facial landmark detection on images

  Detecting facial landmarks is a subset of the shape prediction problem. Given an input image a shape predictor attempts to localize key points of interest along the shape. Detecting facial landmarks is therefore a two step process: Step 1: Localize the face in the image. Step 2: Detect the key facial Structures on the face ROI.

- Facial landmark detection on input video stream

  First up all to initialize dlibs HOG+Linear SVM object detector for face detection and then load the facial landmark predictor. Setting up the video stream pointers and then polling the stream for frames. Detecting facial landmarks in a single image then detecting facial landmarks in a series of frames.

- Extraction of eye regions

  Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye , and then working clockwise around the remainder of the region:

  where p 1 , . . . , p 6 are the 2D landmark locations, de- picted in Fig.

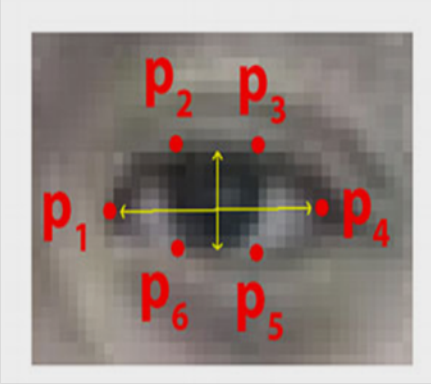

Figure 5.4: Extraction of eye regions

- Computing the eye aspect ratio(EAR)

  The EAR is mostly constant when an eye is open and is getting close to zero while closing an eye. It is partially person and head pose insensitive. Aspect ratio of the open eye has a small variance among indi- viduals and it is fully invariant to a uniform scaling of the image and in-plane rotation of the face. Since eye blinking is performed

by both eyes synchronously, the EAR of both eyes is averaged.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 5.5: Eye Aspect Ratio

1.when eye is open
2.when eye is closed

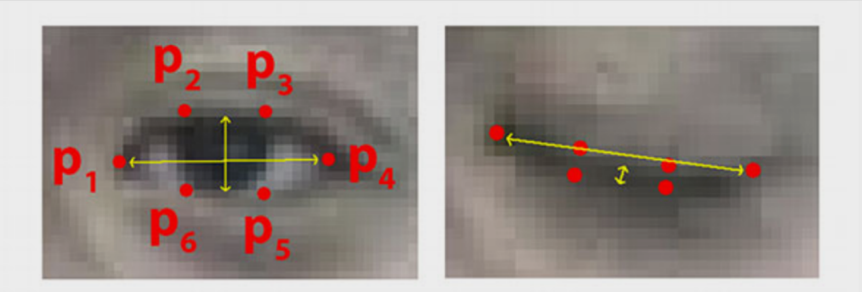

Figure 5.6: Opening and closing of eye

- Counting the number of blinks

  When determining if a blink is taking place in a video stream, we need to calculate the eye aspect ratio. If the eye aspect ratio falls below a certain threshold and then rises above the threshold, then it is register as blink.Defaultly take the value 0.3. If three successive frames with an eye aspect ratio less than a value 3 must happen in order for a blink to be registered.

- Sound an alarm when drowsiness detected

  If the total number of consecutive frames aspect ratio is exceeds a threshold value then we assume the person is starting to doze off.Then playing the alarm sound to alert the driver.

5.4 Input Output Design

- Facial landmark detection on images

Input: Image
Output: Facial landmark on image

- Facial landmark detection on input video stream

  Input: Real time input video stream
  Output: Facial landmark in real time

- Extraction of eye regions

  Input: Real time input video stream
  Output: Facial landmark with only eye region

- Counting the number of blinks
  Input: Real time input video stream
  Output: Shows the total number of blinks

- Sound an alarm

  Input: Real time input video stream
  Output: Alert using an alarm when the eye is closed

Chapter 6

System Implementation

A real-time algorithm to detect eye blinks in a video sequence from a standard camera is implemented in the project. algorithm therefore estimates the landmark positions, extracts a single scalar quantity eye aspect ratio (EAR) characterizing the eye opening in each frame. Finally, an SVM classifier detects eye blinks as a pattern of EAR values in a short temporal window. The simple algorithm outperforms the state-of-the-art results on two standard datasets.

6.1 Platform and Tools

The development of this project was done in Macintosh environment. It can run on Linux, Windows and Windows systems. We use Xcode as the code editor during development.

Platform

System - Macintosh/Linux is preferred for its user friendliness and vast set of available tools. It provides an easy way to code and test the algorithm along with i/o support

Webcam - Logitech C920. It is relatively affordable.It can shoot in full 1080p.It is plug-and-play compatible with nearly every device Ive tried it with (including the Raspberry Pi).

Homebrew - Homebrew installs the stuff you need that Apple (or your Linux system) didnt.Homebrew installs packages to their own directory and then symlinks their files into /usr/local.Homebrew wont install files outside its prefix and you can place a Homebrew installation wherever you like.Trivially create your own Homebrew packages.

Tools

The algorithms for facial landmark detection in image,facial landmark detection in video stream,extraction of eye region,blink calculation and EAR calculation ,Drowsiness detection are all implemented using Python and . Python provides extensive set of libraries and packages for Image processing functionalities which enabled us to develop the project with great ease.

Python libraries included are:-

- dlib
  Developed by Davis King, the dlib C++ library is a cross-platform package for threading, networking, numerical operations, machine learning, computer vision, and compression, placing a strong emphasis on extremely high-quality and portable code. The documentation for dlib is also quite fantastic. From a computer vision perspective, dlib has a number of state-of-the-art implementations, including:
  1.Facial landmark detection
  2.Correlation tracking
  3.Deep metric learning

  - Boost: Boost is a collection of peer-reviewed (i.e., very high quality) C++ libraries that help programmers not get caught up in reinventing the wheel. Boost provides implementations for linear algebra, multithreading, basic image processing, and unit testing, just to name a few.

  - Boost.Python: As the name of this library suggests, Boost.Python provides interoperability between the C++ and Python programming language.

  - CMake: CMake is an open-source, cross-platform set of tools used to build, test, and package software. You might already be familiar with CMake if you have used it to compile OpenCV on your system.

  - X11/XQuartx: Short for X Window System, X11 provides a basic framework for GUI development, common on Unix-like operating systems. The macOS/OSX version of X11 is called XQuartz.

- OpenCV
  OpenCV was designed for image processing. Every function and data structure has been designed with an Image Processing application in mind. In real time image processing with both Matlab and OpenCV, we usually got very low speeds, a maximum of about 4-5 frames being processed per second with Matlab. With OpenCV however, we get actual real time processing at around 30 frames being processed per second. Matlab uses just way too much system resources. With OpenCV, we can get away with as little as 10mb RAM for a real-time application. Thus OpenCV is a better choice than

Matlab for a real-time drowsiness detection system.

dlib C++ library is a cross-platform package for threading, networking, numerical operations,machine learning, computer vision, and compression, placing a strong emphasis on extremely high-quality and portable code. The documentation for dlib is also quite fantastic.From a computer vision perspective, dlib has a number of state-of-the-art implementations, including:
1.Facial landmark detection
2.Correlation tracking
3.Deep metric learning

6.2 Sample Code

Sample Python File

In here we define the analysis process

```
    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 < EYE_AR_THRESH:
        COUNTER += 1


        if COUNTER >= EYE_AR_CONSEC_FRAMES:

            if not ALARM_ON:
                ALARM_ON = True


                if args["alarm"] != "":
                    t = Thread(target=sound_alarm,
                        args=(args["alarm"],))
                    t.deamon = True
                    t.start()


            cv2.putText(frame, "You are sleepy dear take rest", (10, 30),
                cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


    else:
        COUNTER = 0
        ALARM_ON = False



cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
```

Figure 6.1: Sample code for Python

Drowsiness detection

Drowsiness is detected when the driver is sleepy

– Facial landmark detection on images

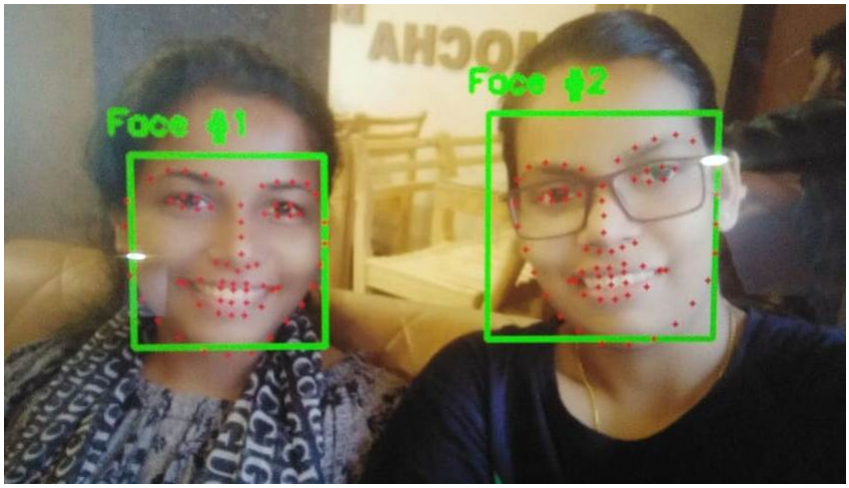

Figure 6.2: Facial landmark on images

– Facial landmark detection on videostream

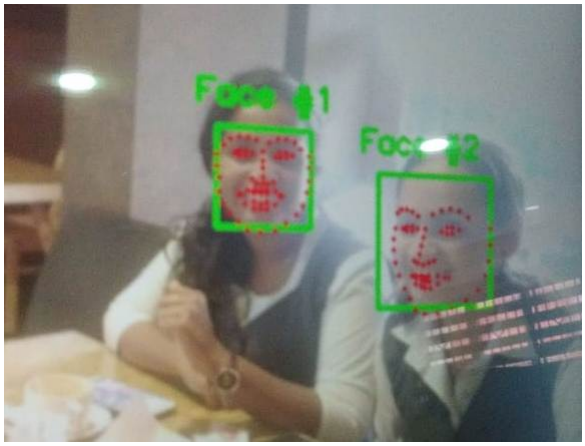

Figure 6.3: Facial landmark on video

– Extraction of eye regions

Figure 6.4: Extraction of eye

– Computing the eye aspect ratio(EAR)

Computing EAR by using the following equation.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 6.5: EAR

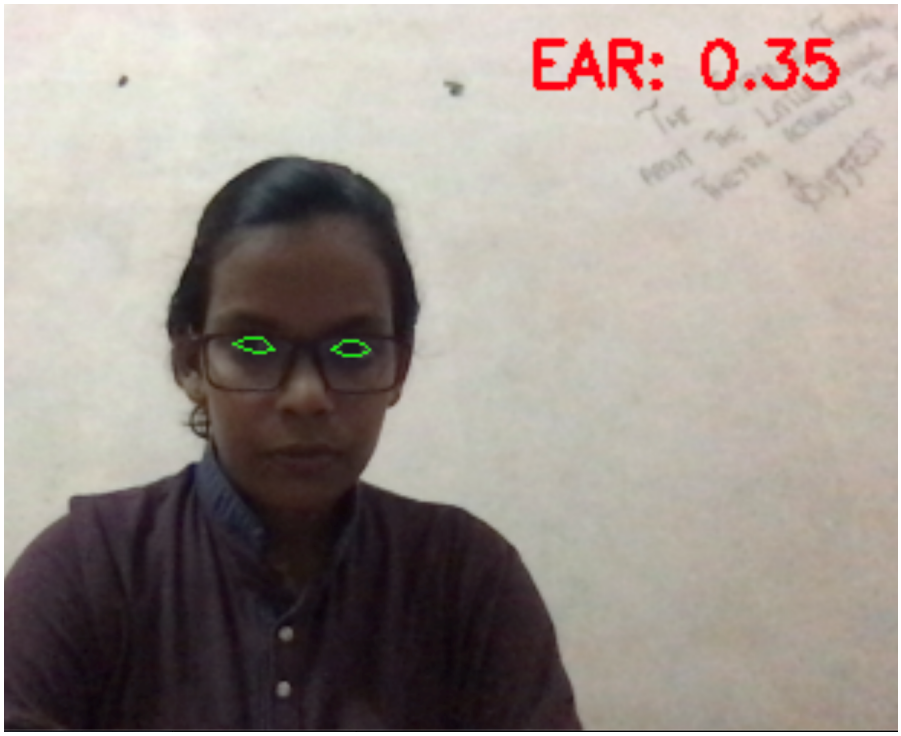

Figure 6.6: EAR calculation

– Counting the number of blinks

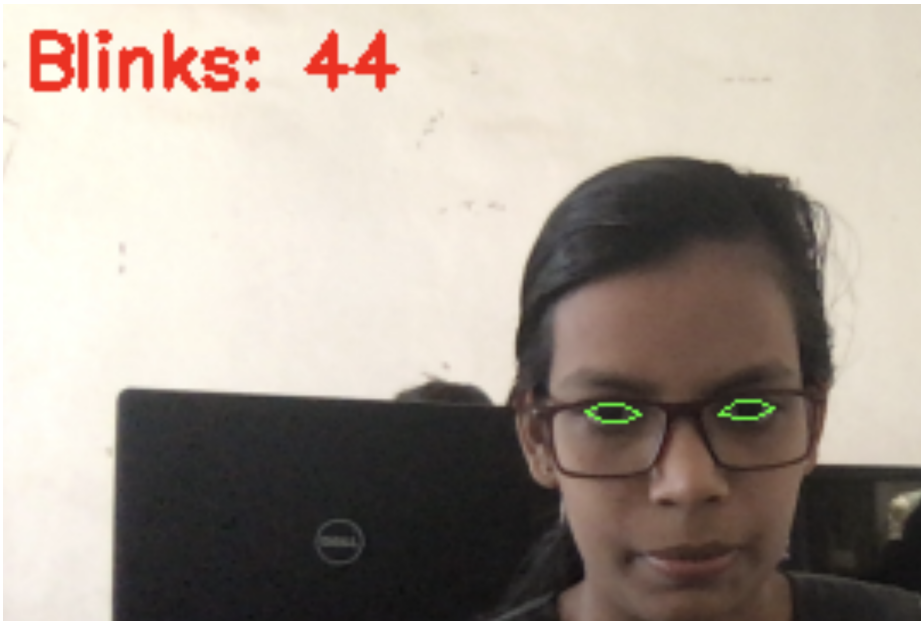

Figure 6.7: Blink count

– Sound an alarm

Figure 6.8: Drowsiness detection

Chapter 7

Result

The proposed system can prevent the accidents due to the sleepiness while driving. The system works well even in case of drivers wearing spectacles and even under low light conditions if the camera delivers better output. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for too long, a warning signal is issued. processing judges the drivers alertness level on the basis of continuous eye closures.

Chapter 8

Future Scope

In future,this prototype can be extended to give alarm before sleeping by calculating the heart beat measure without physical disturbance i.e., non intrusive method using modified ECG methods. Usually in ECG method key points of body For example chest, head, wrist etc , are sticked with wire. In the extended method, sticking wire may be avoided. This will lead us to a way to find out the optimum level of drowsiness.

Chapter 9

Conclusion

Building the drowsiness detector using OpenCV, dlib, and Python. Our drowsiness detector hinged on two important computer vision techniques: (i) Facial landmark detection (ii) Eye aspect ratio Facial landmark prediction is the process of localizing key facial structures on a face, including the eyes, eyebrows, nose, mouth, and jawline. Specifically, in the context of drowsiness detection, we only needed the eye regions . The proposed system in this analysis provides accurate detection of driver fatigue. The analysis and design of driver drowsiness detection system is presented. The proposed system is used to avoid various road accidents caused by drowsy driving and it can also help drivers to stay awake when driving by giving a warning when the driver is sleepy. And also this system used for security purpose of a driver. During the monitoring, the system is able to decide if the eyes are opened or closed.

Chapter 10

Reference

Books:

[1] One millisecond face alignment with an ensemble of regression trees Vahid Kazemi, Josephine Sullivan,Published in IEEE Conference on Computer Vision and Pattern 2014.

[2] Real-Time Eye Blink Detection Using Facial Landmarks.Tereza Soukupova and Jan Cech ,21st Computer Vision Winter Workshop

[3]A. Asthana, S. Zafeoriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In Confer- ence on Computer Vision and Pattern Recognition, 2014.

[4]J. Cech, V. Franc, and J. Matas. A 3D approach to facial landmarks: Detection, refinement, and track- ing. In Proc. International Conference on Pattern Recognition, 2014.. M. Chau and M. Betke. Real time eye tracking