Internet of Things (IoT) Project Report

**on**

**Driver Drowsiness Detection System**

Project Report Submitted

in Partial Fulfillment of the Requirements for the Degree of

**Bachelor of Technology**

**in**

**Computer Science Engineering/Computer Science**



## SCHOOL OF ENGINEERING AND TECHNOLOGY

BML MUNJAL UNIVERSITY GURGAON

by

**Submitted to:**

Dr. Goldie Gabrani

School of Engineering and Technology

BML Munjal University

**Submitted by:**

Shivam Tripathi (1800268C203)

Jai Kumar (1800223C203)

Lakshay Pahuja (1800231C203)

# Table of Contents

|  |  |  |
| --- | --- | --- |
| Serial No. | Topic | Page(s) |
| 1 | **Problem Statement** | **3** |
| 2 | **Explanation** | **3** |
| 3 | **Abstract** | **3** |
| 4 | **Motivation** | **3** |
| 5 | **Specific Problem Solved** | **4** |
| 7 | **Block Diagram** | **5** |
| 8 | **Flow Chart** | **6** |
| 9 | **Hardware Requirements** | **6** |
| 10 | **Software Diagram** | **7** |
| 11 | **Detailed Solution** | **7** |
| 12 | **Existing State-of-the-art** | **8** |
| 13 | **Similar other approaches** | **9** |
| 14 | **Our approach** | **9** |
| 15 | **Idea first conceived** | **10** |
| 16 | **Additional Information** | **10** |
| 17 | **References** | **11** |

# Problem statement and Explanation

**To detect the driver drowsiness for avoiding accidents by using advanced Image Processing.**

Drowsiness of a person can be measured by the extended period of time for which his/her eyes are in closed state. In our system, primary attention is given to the faster detection and processing of data.

The number of frames for which eyes are closed is monitored. If the number of frames exceeds a certain value, then a warning message is generated on the display showing that the driver is feeling drowsy.

In our algorithm, first the image is acquired by the webcam for processing. Then we use the Haarcascade file face.xml to 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 by using Haarcascade\_eye.xml.

If an eye is detected then there is no blink and the blink counter K 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.

# 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 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 Haarcascade library for the detection of facial features.

# Our Motivation

Driver drowsiness is a serious hazard in transportation systems. It has been identified as a direct or contributing cause of road accident. Driver drowsiness is one of the major causes of road accident. Drowsiness can seriously slow reaction time, decrease awareness and impair a driver's judgment. It is concluded that driving while drowsy is similar to driving under the influence of alcohol or drugs. In industrialized countries, drowsiness has been estimated to be involved in 2% to 23% of all crashes. Systems that detect when drivers are becoming drowsy and sound a warning promise to be a valuable aid in preventing accidents. Possible techniques 3 for detecting drowsiness in drivers can be generally divided into the following categories: sensing of physiological characteristics, sensing of driver operation, sensing of vehicle response, monitoring the response of driver. Driver fatigue is a significant factor in a large number of vehicle accidents. Recent statistics estimate that annually 1,200 deaths and 76,000 injuries can be attributed to fatigue related crashes. The development of technologies for detecting or preventing drowsiness at the wheel is a major challenge in the field of accident avoidance systems. Because of the hazard that drowsiness presents on the road, methods need to be developed for counteracting its affects.

# Specific Problem Solved

There are many signs by which we can detect the drowsiness of drivers. They are as given below: − Driver may be yawn frequently. − Driver is unable to keep eyes open. − Driver catches him nodding off and has trouble keeping head up. Driver drowsiness detection system is a system that is implemented using image processing to detect drowsiness of the driver. This application can be very useful to reduce the accidents, because most accidents occurs due to drowsiness of drivers. With use of this application the driver’s status can be monitored, like yawning, fatigue, closure of eyes etc. In this system, the camera is placed in front of driver. The status of driver’s face is continuously monitored using the camera. The alarm can be used to give an alert, if any sign of drowsiness is detected. The proposed system will work in three main stages. 1) In first stage the face of the driver is detected and tracked. 2) In the second stage the facial features are extracted for further processing. 3) In last stage, eye’s status is monitored. In this last stage it is determined that whether the eyes are closed or open. On the basis of this result the warning is issued to the driver. For this system I am going to use Raspberry Pi with raspbian (linux) OS. The camera will be connected through USB port of raspberry pi. The image processing will be done using OpenCV.

We developed a simple system consisting of 5 modules namely (a) video acquisition, (b) dividing into frames, (c) face detection, (d) eye detection, and (e)drowsiness detection. Each of these components can be implemented independently thus providing a way to structure them based on the requirements. Four features that make our system different from existing ones are: (a) Focus on the driver, which is a direct way of detecting the drowsiness (b) A real-time system that detects face, blink, and driver drowsiness (c) A completely non-intrusive system, and (d) Cost effective

# Detailed Solution

We propose an ESD (Eyelid’s State Detecting) value, which is a measurement used to classify the state of eyelid, open or close. The value can be computed by using the algorithm shown in Figure 7. The objective of the algorithm is to find the minimum threshold, which brings the binary image having at least one black pixel after applying median blur filtering. In this algorithm, we use a half-bottom eye image from the selected area by the previous algorithm. We then threshold the image with the threshold value (begin with 0). After that, we apply a median blur filter to the threshold image and check whether at least one black pixel appears. If there is no black pixel, we increase the threshold value and follow the same sequence, but if there is more than one black pixel, we terminate the process and get the ESD value as that threshold. For a faster computation, a binary search implementation is suggested.

Figure 8 shows example images of two states of eyelid, open and close. The threshold images and the median blur images of the open state and the close state are shown in Figure 9 and Figure 10, respectively. The first row images present the threshold images and the second row presents the corresponding blur images. The ESD value of the presented open state is between 10 and 20, which is 18, and the ESD value of the presented close state is between 30 and 40, which is 36.



Fig 3.2.8.1: The open state and the close state of eyelid

A picture containing calendar

Description automatically generated

Fig 3.2.8.2: The threshold images (above) and the corresponding median blur images (below) of the open state.

A picture containing table

Description automatically generated

Fig 3.2.8.3 The threshold images (above) and the corresponding median blur images (below) of the close state.

In our drowsiness detector case, we’ll be monitoring the eye aspect ratio to see if the value falls but does not increase again, thus implying that the driver/user has closed their eyes.

Once implemented, our algorithm will start by localizing the facial landmarks on extracting the eye regions.

We can then monitor the eye aspect ratio to determine if the eyes are closed.

And then finally raising an alarm if the eye aspect ratio is below a pre-defined threshold for a sufficiently long amount of time (indicating that the driver/user is tired):

In the next section, we’ll implement the optimized drowsiness detection algorithm detailed above on the Raspberry Pi using OpenCV, dlib, and Python.

Link to research paper: <https://drive.google.com/file/d/1nYBOm3haztFP8AqZG1-I19ijkYBE7IUV/view?usp=sharing>

# Existing State-of-the-art

OpenCV

OpenCV was designed for image processing. Every function and data structure has been designed with an Image Processing application in mind. Meanwhile, Python is a general purpose programming language started by ***Guido van Rossum***, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability. Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works; it is a Python wrapper around original C++ implementation. And the support of Numpy makes the task easier. **Numpy** is a highly optimized library for numerical operations. It gives MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

Machine Learning

The goal of machine learning is to turn data into information. After having learned from a gathering of data, we want a machine that is able to answer any question about the data:

• What are the other data that are similar to given data? Is there a face in the image?

• What kind of ad will influence the user?

• There is usually a cost component, so the question may become:

Of the many products that we can make the money from, which one will most likely be bought by the user if an ad is shown for it? Machine learning converts data into information by detecting rules or patterns from that data.

# Similar other approaches and solutions

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Existing state-of-art | Drawbacks | Overcome |
| 1 | Yawn Detection with alarm based system | May shock driver and lead to accident | Refer to our approach |
| 2 | Yawn detection using water sprinkler | Less grip on steering may lead to accident |
| 3 | Yawn detection with Ejecto Seato | Vigorous movement of seat may lead to accident |

# Our approach

Name

Yawn Detection with Seat Vibration Motor System

USP

Different from other available approaches as it is a safer way to alert the driver.

In our case, instead of alerting the driver using a high pitch alarm sound or a water sprinkler, we will be setting up a vibrating motor that does not catch the driver off guard while driving as it may lead to an accident.

# Technical Features

Face Detection

In this stage we detect the region containing the face of the driver. 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 type of technology with the use of computer. The frame may be any random frame. Only facial related structures or features are detected and all others types of objects like buildings, tree, bodies are ignored.

Eye Detection

After successful detection of face eye needs to be detected for further processing.

In our method eye is the decision parameter for finding the state of driver. Though detection of eye may be easier to locate, but it’s really quite complicated. At this point it performs the detection of eye in the required particular region with the use of detection of several features. Generally Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver.

State of eye

In this stage, we find the actual state of the eye that if it is closed or open or semi closed or open. The identification of eyes status is most important requirement. It is achieved by an algorithm which will be clarified in the later parts. We channelize a warning message if we obtain that the eyes are in open state or semi open state up to a particular threshold value. If the system detects that the eyes are open then the steps are repeated again and again until it finds a closed eye.

Drowsiness Detection

After getting eyes the algorithm then counts the number of open eyes form each frame and determines the drowsiness. If the criteria are satisfied, then the driver is said to be drowsy. The buzzer connected to the system performs actions to correct the driver abnormal behaviour.

A variable will store the number of successive frames in which the eyes found to be closed with the values like 0, 1, 2, 3... etc. Initially, this variable is set to 0. When both the eyes are open, then Drowsycount will be 0. Drowsycount will increase when Eyestotal < 2. For an eye blink, Drowsycount value be raised by 1. If the eyeblinks in more than 4 frames, i.e. variable count is greater than or equal to 4, then the condition for drowsiness is met and an alarm will be signalled at real time.

# Block Diagram

Diagram

Description automatically generated

Explanation

Drowsiness of a person can be measured by the extended period of time for which his/her eyes are in closed state. In our system, primary attention is given to the faster detection and processing of data.

The number of frames for which eyes are closed is monitored. If the number of frames exceeds a certain value, then a warning message is generated on the display showing that the driver is feeling drowsy.

In our algorithm, first the image is acquired by the webcam for processing. Then we use the Haarcascade file face.xml to 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 by using Haarcascade\_eye.xml.

If an eye is detected then there is no blink and the blink counter K 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.

# Flowchart

**Diagram

Description automatically generated**

**Explanation**

In the proposed system, the raspberry pi and android application works together. The following Flow charts depicts the flow of the whole system where, firstly the driver registers himself, logs in when they are driving. As soon as the system is turned on, it starts detecting the drowsiness of the driver and when detected, an alarm is being invoked, alert message is being sent to the emergency contacts and the owner and also if not detected the system continues to monitor the driver's state. At the same time if an owner wishes to check the current location, he/she can login into the app and can see their current location if they are logged into the system and driving. And also it is used to keep a track of the driver's record with the help of cloud firebase. The flow to the system is being displayed in the following figure.

Hardware requirements

Our system was run in a PC with a configuration of 1.6GHz and 1GB RAM Pentium dual core processor.

Though the system runs fine on higher configurations, when a system has an inferior configuration, the system may not be smooth and drowsiness detection will be slow.

The problem was resolved by using dedicated hardware in real time applications, so there are no issues of frame buffering or slower detection.

# Software Requirements

* Python:
  + Python 3 3.2 Libraries
  + Numpy
  + Scipy
  + Playsound
  + Dlib
  + Imutils
  + opencv, etc.
* Operating System
  + Windows 7 or above
  + Debian
  + macOS X or 11

# New Features

We have proposed our system to overcome the flaws of the existing system. Our system has features like real-time drowsiness detection using eye detection, yawning detection, and vehicle tracking. Our system is named Driver Awakening System. Our system comprises of hardware parts and the software parts. Our system’s hardware part is installed in the vehicle and the software part consists of an application named Driver Assistant. The hardware part consists of a Raspberry pi and web camera. When the vehicle gets started our installed system starts working. The driver and the owner have to register first in the Driver assistant application with the required details.

The driver has to login to the application before starting the journey and then the notification would be sent to the owner about who is driving the vehicle along with the vehicle details and there starting location would also be stored in the database. When the vehicle is started the installed system’s camera starts capturing the real-time video. This video will be the input to the raspberry pi which then converts video into the video frames and then these frames are compared to check whether the driver is in the drowsy state or not. The alarm will be invoked if the driver is in fatigue state.

If the number of times the alarm invokes crosses the threshold value then the SMS would be sent to the relative’s and the owner. The driver will feed the relatives contact during registration. If the drowsiness is not detected then the system will continue the whole process. For drowsiness detection algorithms like HaarCascade is been used. The owner can check the real-time position of the vehicle by login to the Driver Assistant application. The Owner can also outlook the details of the driver. This is the proposed system by our group and can further be extended with many updates.

# Alternative Implementation

Instead of using a device, we can also develop a mobile application for the same.

Moving towards the Android Application's role, here we are using and android application as it is a very convenient way of interfacing the database to any physical system. Here, it is given the name "Driver Assistant" as it acts as an assistant to the driver by taking the driver's details, allowing them to login and feed the source and the destination location, allow access to google map and keep a record of them for the owner.

DRIVER: The driver is able to register himself with the help of his contact number and by filling the personal, address, license, vehicle details, emergency contact details. After registering themselves they can login with the help of their contact number with an OTP generated by the API through the application. They can also be able to navigate to the destination by accessing to the google map through the application.

OWNER: The owner is able to login to the application same as the driver i.e., through OTP generation. The owner after logging in is able to check the driver's profile and current location of them while driving. The owner is also able to access to the record of the drivers- combined and individually.

# Status of Project

Graphical user interface

Description automatically generated

In development

# Details

The raspberry pi 3B features dual band IEEE 802 0.11, Wi-Fi, Bluetooth 4.1 and megabit Ethernet.

Specifications:

* Quad core 1.2 gHz Broadcom BCM 2837 64-bit CPU
* 1GB RAM
* 100 base ethernet
* 40 pin extended GPO
* 4 USB 2 ports
* 4 pole stereo output
* Full size HDMI
* CSI camera port for loading your operating system and storing data
* Upgraded switched micro USB source up to 2.5A

# Idea First Conceived

In recent years, driver drowsiness has been one of the major causes of road accidents and can lead to severe physical injuries, deaths, and significant economic losses. Statistics indicate the need of a reliable driver drowsiness detection system which could alert the driver before a mishap happens. Researchers have attempted to determine driver drowsiness using the following measures:

* vehicle-based measures
* behavioural measures
* physiological measures

A detailed review on these measures will provide insight on the present systems, issues associated with them and the enhancements that need to be done to make a robust system. In this paper, we review these three measures as to the sensors used and discuss the advantages and limitations of each. The various ways through which drowsiness has been experimentally manipulated is also discussed. We conclude that by designing a hybrid drowsiness detection system that combines non-intrusive physiological measures with other measures one would accurately determine the drowsiness level of a driver. A number of road accidents might then be avoided if an alert is sent to a driver that is deemed drowsy.

# Additional Information

Speed The speed of system is defined as the frequency of total program instructions for one particular cycles. Higher the frequency, better the performance.

Speed = cycles per second

Speed = 1 / Tm

Where, Tm is time taken by one complete cycle of total program instructions.

Driver Drowsiness Detection with Audio-Visual Warning (IJIRST/ Volume 3 / Issue 01/ 048)

Accuracy: The accuracy is defined by the ratio of correctly detected eye state, by the total samples taken by camera. One particular cycle can be explained as one sample.

Accuracy = Nc / Nt

Where, Nc = number of samples in which eye state is detected correctly Nt = Total number of taken samples

Limitations

1. Dependence on ambient light:- With poor lighting conditions even though face is easily detected, sometimes the system is unable to detect the eyes. So it gives an erroneous result which must be taken care of. In real time scenario infrared backlights should be used to avoid poor lighting conditions.

2. Optimum range required:- when the distance between face and webcam is not at optimum range then certain problems are arising.

When face is too close to webcam(less than 30 cm) , then the system is unable to detect the face from the image. So it only shows the video as output as algorithm is designed so as to detect eyes from the face region.

This can be resolved by detecting eyes directly using haardetectobjects functions from the complete image instead of the face region. So eyes can be monitored even if faces are not detected.

When face is away from the webcam(more than 70cm) then the backlight is insufficient to illuminate the face properly. So eyes are not detected with high accuracy which shows error in detection of drowsiness.

This issue is not seriously taken into account as in real time scenario the distance between drivers face and webcam doesn’t exceed 50cm. so the problem never arises.

Considering the above difficulties, the optimum distance range for drowsiness detection is set to 40-70 cm

# Conclusion

Driver drowsiness detection is designed mainly to keep the driver awake while driving to avoid the accident due to sleepiness. The alert signal is generated from embedded device to awake driver from sleepy state. The Pi along with Raspbian camera is used to calculate the drowsiness of the driver in real time. Fatigue is measured by detecting Eye and face using HaarCascade Classifier, especially facial landmarks is detected using shape-predictor and Eye Aspect Ratio (EAR) by calculating the Euclidean distance between the eyes. Accurate eye detection and faces in every frame will help to calculate drowsiness level. Frequent detection of eye blinking and head tilting is measured properly and it helps to indicates drowsiness. When he/she reaches maximum threshold the driver will be alarmed by a loud warning that will wake up the driver from the sleep state. In future, the implementation can be carried out in a bright room with consistent light, for different lighting conditions and also for the persons with dark skin can be considered.

# References

1. L. Jia, D. Zhao, K. Zheng, Z. Li, G. Sun, and F. Zhang, “Smartphone-based fatigue detection system using the progressive locating method”, IET Intell. Transp. Syst, vol. 10, no. 3, pp. 148–156, 2016
2. J. S. Jayasenan and P. S. Smitha, “Driver Drowsiness Detection System”, vol. 4, no. 1, pp. 34–37, 2014
3. Mayank Chauhan, Mukesh Sakle, ”Study & Analysis of Different Face Detection Techniqes” International Journal of Computer Science and Information Technologies, Vol. 5 (2) , pp 1615-1618,2014.
4. [4]Paul Viola and Michael j. Jones,” Rapid Object Detection using a Boosted Cascade of Simple Features,” International Journal of Computer Vision 57(2), pp 137–154, 2001.