
Implementing a driver drowsiness detection system

Athena Karumbaya
akanjith@andrew.cmu.edu

Dhanashree Balaram
dbalaram@andrew.cmu.edu

Abstract

More than 1.25 million people die each year as a result of road accidents which accounts for 2.2% of deaths annually [5]. This rate can be significantly reduced if the driver is notified when he or she is unfit to drive. This paper proposes a solution to mitigate vehicular related accidents by detecting the capability of a person to drive. Information from various sensors measuring heart rate, stress levels, perspiration and drowsiness are combined using Machine Learning to derive crucial information about the state of the driver and decide if he or she is in a fit condition to drive.

1 Introduction

Statistics indicate that road accidents are one of the major reasons for increase in deaths in young adults between the ages of 15 to 44 [5]. A significant amount of these road accidents are caused by drowsy drivers. The National Highway Traffic Safety Administration estimates that drowsy driving was responsible for 72,000 crashes, 44,000 injuries, and 800 deaths in 2013. These numbers estimate that up to 6,000 fatal crashes each year [6]. There is a 12.5 billion in monetary losses each year due to driver fatigue. According to a survey [6] nearly 150,00 adults in among 19 states in the District of Columbia have accepted to falling asleep on the wheel.

With the booming growth of technologies such as internet of things and machine learning, there is an increase in trend to incorporate the use of sensor networks and artificial intelligence to reduce the number of vehicular accidents caused annually. Through the course of this paper, we will introduce some existing products that pose as our competition. We will also discuss in detail our approach to solving driver drowsiness. We performed a number of experiments which will be listed in section 5.

2 Related Work

In the World of automation today, there is a lot of ongoing research on monitoring driver fatigue and drowsiness. Some of these have made it into production and are being sold as individual units in the consumer market. These research include incorporating sensors into cars to monitor the driver. Driver capability can be detected by integrating a device with the car infotainment system, or it could be a wearable device.

Shenzhen Loyal Goal Industry Electronic Co.Ltd has developed a 'Driver fatigue alarm' that looks like a Bluetooth earpiece. It reacts to head movements and sends a gentle vibration or a customized sound as soon as it detects the lowering of the head when the driver becomes sleepy.

Bosch has developed a camera-based system that will monitor head and eye movements, as well as body posture, heart rate and body temperature. It detects for early signs of fatigue and advises drivers to take a break. While this product is still in research stage, Bosch has another product to detect driver drowsiness by learning the pattern of how he/she holds the steering.[1]

'Plessey Semiconductors' in collaboration with Nvidia has developed technology that can detect when the driver becomes sleepy by using electric potential sensor system. The technology can be used to detect heart beats and other nerve and muscle activity with or without dry contact by measuring changes in an electric field.[4]

The company 'Optalert' developed drowsiness detection glasses that measure the velocity of the operator's eyelid 500 times a second using a tiny invisible LED built into the frame of the glasses. It measures how fast and how far a person opens their eyelid after they close it. [3] 'Guardian Systems' Drowsiness system for truck drivers uses a forward facing camera to detect face and eye movements to detect early signs of driver fatigue.

3 Approach

The proposed approach has three stages. The first one is to collect sensor data from a pulse sensor(heart rate), a galvanic skin response(stress-level) sensor as well as collect visual data from the RasPi (picamera) sensor. The next stage is to preprocess the visual information(pictures) and predict whether the eyes of the driver are open or closed. The output of this stage and the numeric outputs from the sensors are assimilated and used to make a decision about whether a driver is capable or not.

3.1 Processing information from Sensors

Pre-processing is only carried out for the images that are collected with the camera sensor. The other two sensors provide us with raw sensor readings that are used as our dataset. These readings can be visualized in real time on an online cloud dashboard called Ubidots as shown in Figure 1



Figure 1: Dashboard visualizing sensor information

3.2 Pre-processing images

The picamera apparatus connected to the Raspberry pi captures a fixed number of video frames of the driver before processing. These photographs are segmented and a Haar Cascade classifier is used to learn the region of interest; the eyes. The part of the image that contains the eyes is reshaped to a size of 24x24x1 (gray scale) pixels.

3.3 Training the Convolutional Neural Network(CNN)

The primary task of training is to find a suitable dataset. In this project we used an open source dataset for eyes known as 'Closed Eyes in the Wild dataset'. [2] In order to learn the features of closed and open eyes, we train a deep convolutional neural network with max pooling, ReLU activation and dropout. The optimization used is stochastic gradient descent. The training dataset consists of 3876 samples of both open and closed eyes of size 24x24x1.

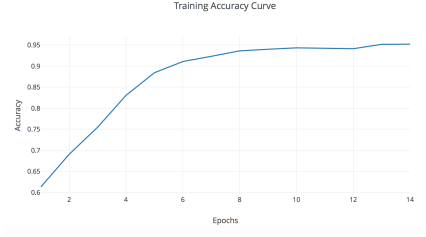


Figure 2: Train Accuracy Curve

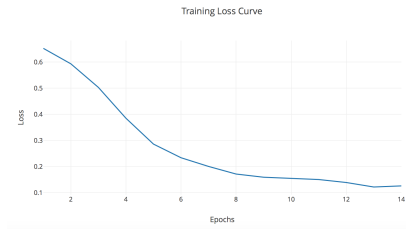


Figure 3: Train Loss Curve

3.4 Training the Decision Tree

Once we have all three of the inputs, namely, heart rate(beats per minute), skin conductance sensor reading and decision of open/closed eyes, it is possible to use thresholding and generate our own dataset that simulates the output as to whether the driver is capable or not. Then, using a decision tree structure, a pattern that traces the dataset is recognized. Each input attribute is very important for information gain. The decision tree is trained using the 'xgboost' package. The inputs are integers and the output is binary. The trained decision tree is ultimately a binary classifier that outputs '0' if the driver is incapable and '1' if the driver is capable.

3.5 Running inference

Now, when we have a test image. The camera captures 10 continuous frames. The eyes that are recognized in each image are extracted and stored. After this, each image is passed through the weights learnt in the CNN and the output is binary, i.e 0 or 1 (closed or open), for each image. We use some novel approaches to eliminate blinking and light sensitivity. We perform a majority wins on all the frames and output 0/1 as a collective output from the image system. Now this information combined with sensor data goes through the decision tree prediction stage, and a conclusion is drawn as to whether the driver is capable or not. If the output is 'incapable', then an alert is sent to the emergency contact of the driver. In order to wake the driver up a buzzer is also switched on for a period of 3 seconds.

4 Experimental Setup

The architecture for our approach is as shown in figure 4. The pulse sensor will be used to monitor the beats per minute and galvanic skin response sensor will be used to record the perspiration levels of the driver. These sensors are connected to a particle photon which will send sensor information to an online IoT platform, Ubidots. The advantage of these sensors is that they can both run on a constant voltage supply of 3.3V. This can be supplied by the particle photon directly without the need for a step-up/step -down converter or an external power source. These sensors will be attached onto the steering wheel of the car such that the a user is continuously monitored while he or she is driving. The entire apparatus can be fitted into a car so the type of car is not a restriction.

Ubidots is an online IoT platform that is designed for integrating various embedded devices to enable communication, shared information and actuation through specifying thresholds. We have selected this platform to send all the sensor information to the cloud to visualize sensor graphs and driver status in real-time. Particle photon is used as the hardware unit that collects sensor information because it is small, light weight and consumes less power. It has an inbuilt Wi-Fi module which connects it directly to a cloud platform. It also has GPIO pins to connect to sensors and to serially read information.

On the dashboard, a raspberry pi with an attached front facing picamera module is placed which will be oriented to view the driver's face. Raspberry pi is chosen as the hardware platform for driver eye detection because of its high processing power. It has a quad core processor which supports multi-threading. Moreover, it has a Unix like operating system, Raspbian, that supports high level languages such as python which is required to run the deep learning model. It also has support to plug in the camera module with supported libraries as well as GPIO binaries which can be used to

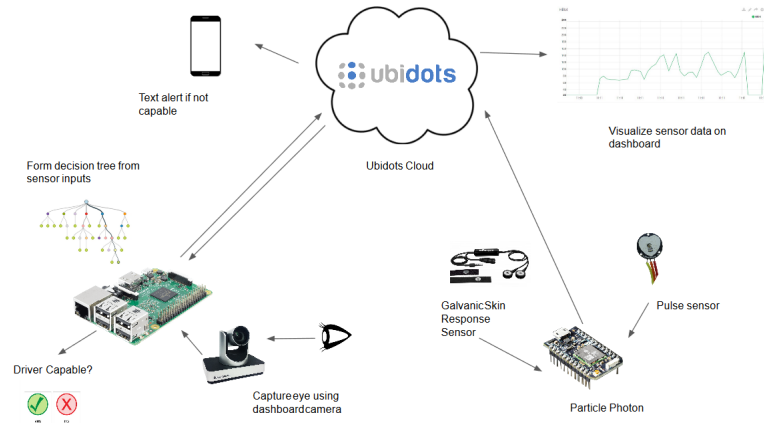


Figure 4: System Architecture

trigger an actuator, which in our case is a buzzer.

The camera sensor detects the driver's eyes and makes a decision about whether they are predominantly opened or closed using a trained deep convolutional neural network (CNN). The CNN output as well as the sensor information pulled from the cloud will enable the system to make a decision whether the driver is fit to drive, using a trained decision tree. This decision is sent back to the Ubidots cloud to view the current status of the driver. If a driver is deemed unfit to drive, a text alert would be sent to the user's mobile device alerting him to be careful. Also, a buzzer will be activated for 3 seconds to alert the driver. (as shown in Figure ??arch)

4.1 Training

The deep CNN is trained for 14 epochs, producing a validation accuracy of 96.29% and a test accuracy of 94.3%. The packages used for this implementation are Keras with Theano backend.

4.2 Testing

An arduous task during the development of any new product is testing it and simulating different conditions for the testing. Emulating a high beats per minute count was done by making the experiment subject (human) run. This would increase the sweat and perspiration as well. Experiments were conducted with different combinations of these readings, as well as the action of the eyelids. These experiments have been consolidated into Figure 5

Beats per Minute	Galvanic Skin Response	Eyes Open/Close	Driver Capability
67	1653	0	Not Capable
71	1789	1	Capable
69	1255	1	Not Capable
38	1782	1	Not Capable

Figure 5: Test Cases

Two test cases with eyes fully open and eyes fully closed have been provided in Figure 6

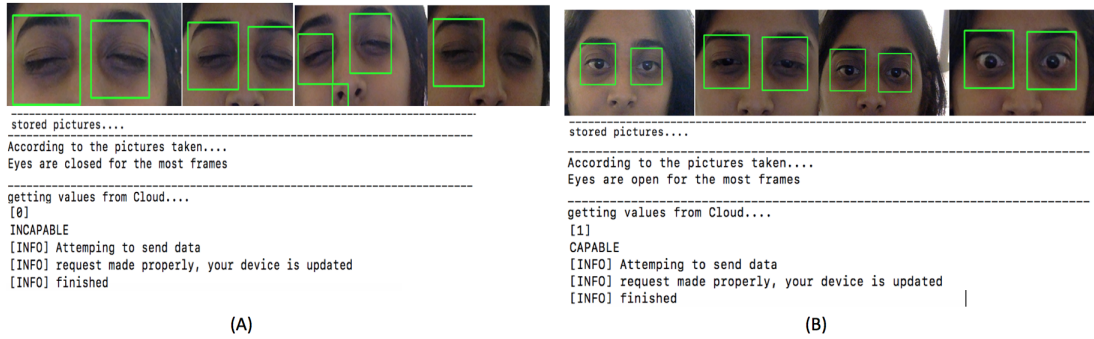


Figure 6: Prediction (A) incapable and (B) capable

5 Results

Having tested on 30 different combinations of values, we achieved a precision of 86.96% , and Accuracy of 71.43% and an FI score of 76.92%. A confusion matrix for all the tests we performed is shown in Figure 7. A picture of the text message sent to the emergency contact is shown in Figure 8

	Capable	Incapable
Capable	20	9
Incapable	3	10

Figure 7: Confusion Matrix; FP = 3, TP = 20, FN = 9, TN = 10



Figure 8: Text message Alert!

References

- [1] Bosch-driver drowsiness. <http://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/driver-assistance-systems/driver-drowsiness-detection/>.
- [2] Closed Eyes In The Wild (CEW). <http://parnec.nuaa.edu.cn/xtan/data/ClosedEyeDatabases.html>.
- [3] OptaAlert drowsiness system. <http://www.optalert.com>.
- [4] Plessey-Nvidia camera sensor system. <http://www.nytimes.com/2017/03/16/automobiles/wheels/drowsy-driving-technology.html>.
- [5] Road crash statistics. <http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics>.
- [6] Statistics for driver drowsiness. <http://www.cdc.gov/features/dsdrowsydriving/index.html>.