AVOID - Accident Prevention of Vehicles by Observing Instantaneous Feeds of Driver Drowsiness

Atul Ramkrishnan
Computer Engineering
K.J. Somaiya College of Engineering, Vidyavihar, Mumbai, India

Akshay Arukandy
Computer Engineering
K.J. Somaiya College of Engineering, Vidyavihar, Mumbai, India

Niraj Gujarathi
Computer Engineering
K.J. Somaiya College of Engineering, Vidyavihar, Mumbai, India

ABSTRACT
Driver fatigue is one of the leading causes of car accidents in the world. The system aims to reduce the number of road accidents by detecting drowsiness and alerting the driver. The purpose of this paper is to develop a driver fatigue detection system. This system uses eye blinking frequency and yawning frequency of the driver to analyze drowsy driver and based on the level of drowsiness system will alert the driver. Driver's facial features are captured by using a camera then this video input is used by system to monitor the driver's eyes to detect early stages of sleep as well as short periods of sleep; for video capturing system uses mobile phone camera making system portable and cost effective. Working of proposed system is based on the driver drowsiness detection method using Haar Cascade frontal face Classifier to detect the frontal facial structure followed by using dlib and Convolutional Neural Networks (CNNs) to extract information from sequence of images (video frames) for predicting driver fatigue.

General Terms
Artificial Intelligence, Image Classification.

Keywords
CNN, dlib, Driver Fatigue Detection, Haar Cascade, eye blinking and yawning frequency.

1. INTRODUCTION
Drowsy driving is a major problem all over the world. Driving while drowsy could be fatal for the driver and to other drivers and pedestrians. Drowsy driving is the dangerous combination of driving and sleepiness or fatigue. This can happen if the driver hasn’t had enough sleep or if the driver has sleep disorders or is on any medications that may cause drowsiness. The National Highway Traffic Safety Administration estimates that drowsy driving was responsible for 72,000 crashes, 44,000 injuries, and 800 deaths in 2013.[1] However, these numbers are underestimated and up to 6,000 fatal crashes each year may be caused by drowsy drivers.[2], [3] It is imperative that drowsy driving is detected and mitigated. Our idea is to create an application that aims to reduce the number of accidents by detecting driver drowsiness. The device would be placed on the dashboard and would continuously monitor the face of the driver looking for signs of drowsiness. The signs that could be considered are the frequency of eye blinks, yawns, etc. On detecting that the driver is drowsy, the device would sound an alarm to alert the driver.

2. LITERATURE SURVEY
The systems which are now readily available or have been implemented are mainly done so only considering the eye blink factor as the main feature, hence lacking precision. Considering only one non-intrusive factor proves ineffective as it is heavily depended upon person to person image rendering. AVOID basically aims to use three factors into consideration - yawning frequency, eye blink frequency and duration namely. It provides necessary alerts to the user in any case of drowsy situation.

2.1 Existing Systems
The systems which currently being used are based upon various factors testing behavioral measures using machine learning techniques, analysis of electroencephalogram and contour detection model. These techniques are being used off late in the drowsiness detection sector but again individually each of these methodologies have its own demerits and hence cannot be deployed very easily.

1.1.1 Behavioral measurement using machine learning techniques
Out here a wide range of techniques like support vector machines, convolutional neural networks and hidden Markov models have been used for the analysis of the same. The main point being development of a drowsiness detection system which provides accurate results in an efficient manner is a big challenge[4].

1.1.2 Analysis of Electroencephalogram
Over here[5] the main problem of recognizing the state of somnolence is studied. As such they have designed a system for automatically detecting drowsiness of a driver at the wheel. One of the tasks performed by analysis of electroencephalogram (EEG) is the problem of recognizing it, characterized by the lower level of attention and also the extension of reaction time to any external stimuli that the user may have to go through with.

1.1.3 Using OpenCV and Contour Detection method
The system designed out here[6] detects real time non-intrusive features of eye blink using OpenCV and yawning using Active Contour Method. The given system easily detects and classifies the facial expressions and provides the necessary output. Result shows that eyes blink detection was more accurate for the driver without eye glasses whereas mouth detection without moustache was found best.
3. PROPOSED SYSTEM

3.1 System Architecture

3.2 Working of the Proposed System

The proposed system can be divided into these mentioned phases:

1.1.4 Feature Extraction
In this step AVOID uses Haar Cascade frontal face Classifier to detect the frontal facial structure. Further, for eye and mouth segmentation dlib is used.

1.1.5 Drowsiness Detection
From the extracted eyes and mouth, AVOID measure the eye blink frequency and yawn frequency in order to classify the driver as drowsy/not drowsy. If the driver is detected as drowsy, an alarm is raised.

3.3 Application Setup
Smartphone placed on the dashboard of the vehicle is being used to capture the real-time video frames. The IP webcam application will start the camera service of that device to capture continuous images from front camera. These video frames are used to measure eye blinking and yawning to monitor level of driver’s drowsiness.

4. METHODOLOGY

4.1 Face detection using Haar cascade
Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images.

4.2 Eyes and mouth segmentation using dlib
Dlib is an open source software library written in C++. It also has a library for Python. Dlib contains machine learning algorithms and tools that can be used in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Moreover, the library is intended to be useful in both research and real world commercial projects and has been carefully designed to make it easy to integrate into a user’s C++ application [8].

4.3 Dlib shape predictor
This object is a tool that takes in an image region containing some object and outputs a set of point locations that define the pose of the object. The classic example of this is human face pose prediction, where you take an image of a human face as input and are expected to identify the locations of important facial landmarks such as the corners of the mouth and eyes, tip of the nose, and so forth [9].

For our application, the facial landmarks as contained in the shape_predictor_68_face_landmarks.dat[10] have been used.
4.4 Convolutional Neural Network

Convolutional Neural Network is a feedforward neural network that contains at least one convolutional layer. CNNs are great at capturing local information (e.g. neighboring pixels in an image). CNNs also reduce the complexity of the model due to parameter sharing resulting in a reduced chance of overfitting and reduced training time. Thus, they are commonly used in image classification.

Four main operations exist in the ConvNet[11]:

1.1.6 Convolution

The convolution step acts as a feature detector. The feature detector is a window of dimensions $n \times n$, where $n$ is any integer $> 0$. Usually $n$ is taken as 3, but it is not uncommon to see feature detectors of size $5 \times 5$ or $7 \times 7$[12].

The following is a brief explanation of how this works

1. You place it over the input image beginning from the top-left corner within the borders you see demarcated below, and then you count the number of cells in which the feature detector matches the input image.
2. The number of matching cells is then inserted in the top-left cell of the feature map.
3. You then move the filter by a distance of one stride, where the stride is decided earlier. If you want to capture all the information, the stride should be set to one[12].
4. After doing this to the entire row, you move down and do the same procedure until all the cells have been covered.

![Figure 3. Convolution Layer](image)

1.1.7 Pooling or Sub Sampling

It is common to embed a pooling layer periodically in a ConvNet architecture between consecutive Conv layers. Its function is to gradually reduce the representation's spatial size in order to reduce the number of parameters and computation in the network and thus also to control overfitting. The pooling layer operates on each depth slice of the input individually and resizes it spatially, using the MaxPool operation. The most common MaxPool layer is a filter of size $2 \times 2$ with a stride of 2. The result is a reduction of 75% from the input to the output[13].

1.1.8 Non-linearity

A feedforward neural network without non-linear activation functions is equivalent to a linear neural network without hidden layers. This requires us to introduce a non-linear activation function. The usual candidates are the sigmoid, tanh, and ReLU function. ReLU is the abbreviation for Rectified Linear Unit.

ReLU is an operation working element-wise i.e. is applied per pixel and substitutes every negative pixel value by 0 in the feature map. It serves the purpose of introducing non-linearity in ConvNet, because the maximum real-life data needed to be fed into our ConvNet is non-linear[11].

In practice ReLU is preferred over tanh or sigmoid as it alleviates the problem of vanishing gradient. A custom activation function has used here though:

1. ELiSH: The Exponential linear Squashing Activation Function known as the ELiSH function is one of the most recent AF, proposed by Basirat and Roth, 2018[14]. The ELiSH function is made up of the ELU and Sigmoid functions and it is given by

$$f(x) = \begin{cases} \frac{x}{1+e^{-x}}, & x \geq 0 \\ e^x - 1, & x < 0 \end{cases}$$

2. HardELiSH: The HardELiSH is the hard variant of the ELiSH activation function. The HardELiSH is a multiplication of the HardSigmoid and ELU in the negative part and a multiplication of the Linear and the HardSigmoid in the positive part[14]. The HardELiSH is given by

$$f(x) = \begin{cases} x \times \left(0, \left(1, \left(\frac{x+1}{2}\right)\right)\right), & x \geq 0 \\ (e^x - 1) \times \max \left(0, \left(1, \left(\frac{x+1}{2}\right)\right)\right), & x < 0 \end{cases}$$

![Figure 4. HardELiSH and ELiSH activation function graph](image)

1.1.9 Fully connected layer

The fully connected layer usually constitutes the last few layers of the CNN. As the name suggests, the “Fully Connected” in the fully connected layer indicates that every neuron in the previous layer is connected to every neuron in the next layer.

The convolution layers before the FC layer(s) hold information regarding local features in the input image such as edges, blobs, shapes, etc. Each conv layer hold several filters that represent one of the local features. The FC layer holds composite and aggregated information from all the conv layers that matters the most.

The summation of all neurons in the output layer should sum up to 1. To ensure this, an activation function called as
softmax is usually used. If the output consists of only a single neuron, sigmoid function can also be used instead.

The CNN has been trained on two datasets, Eye dataset from PARNEC (PAttern Recognition and NEural Computing)[15] and YawDD: Yawning Detection Dataset for eyes and yawn [16].

The below graphs show the model loss and accuracy as the CNN is trained over the eye dataset for 25 epochs.

At the end of 25 epoch a validation accuracy of 97.81% is attained.

4.5 Determining the Level of Drowsiness
The system maintains a track of the blinks/yawns. AVOID system is deployed on the android smartphone using the concept of IP camera where camera’s real time video frames is used as input to the trained system model.

As the video from the camera is fed to the system, the first step is the detection of the face. Once the face is recognized, the eyes and the mouth are segmented to be fed into the neural networks. They both are fed into two separate networks which classify them individually based on the thresholds that have been set. If either of these don’t classify as drowsy then it is safe to assume that the driver is not drowsy. Else a level is assigned as per the degree of drowsiness and if that level is high enough, then an alarm is sounded to alert the driver.

4.6 Application Deployment
The driver’s smartphone is being used to capture the real time video frames. Smartphone will be placed on the dashboard of the vehicle is being used to capture the real-time video frames. AVOID uses a IP webcam application which is available on Google Play Store, which will generate local IP address for the driver’s smartphone device which is in the same network. The IP webcam will start camera service on defined port number of that device after that application will provide url to access IP webcam dashboard on browser. AVOID has ‘../shot.jpg’ used the url to get image frames from this dashboard and these image frames are used as real time video feed of driver’s face data. This real time video frames from smartphone’s camera is being used to calculate eye
blinking and yawning frequency to determine whether driver is drowsy or not.

5. TESTING THE SYSTEM
System functionality testing is performed to check whether model is able to classify closed and open eyes also examine whether yawning is detected. Depending on the eye closed duration and yawning duration driver drowsiness detected, the system classifies fatigue into three levels -

Level 0 - In level 0 driver fatigue is determined by means of frequent eye blinks and yawning for the duration of 1-2 seconds. It is easy to be awakened from this level of drowsiness.

Level 1 - Level 1 is a phase where initial stage of sleep is observed. This level of drowsiness last between 3-4 seconds.

Level 2 - In level 2 driver actually falls asleep. This is the critical phase of the system where driver loses consciousness, can be referred as a short nap which last for the duration of 6 seconds.

Here 5 test cases have been considered for each lighting conditions and test result is classified into expected result and actual result. Eye-based drowsiness detection module undergoes the problem of detecting false positives. This may be due to the limitations of smartphone camera which is not able to detect the face region and eye-region.

Table 1. Table captions should be placed above the table

<table>
<thead>
<tr>
<th>Eye closed Duration or yawning duration (in seconds)</th>
<th>Drowsiness level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 2 seconds</td>
<td>Level 0</td>
</tr>
<tr>
<td>3 – 4 seconds</td>
<td>Level 1</td>
</tr>
<tr>
<td>More than 6 seconds</td>
<td>Level 2</td>
</tr>
</tbody>
</table>

Table 2. Eye blink detection module tested for facial appearance factor where in driver is not wearing glasses

<table>
<thead>
<tr>
<th>Lighting conditions</th>
<th>Daytime</th>
<th>Night light</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Result</td>
<td>Expected</td>
<td>Actual Result</td>
</tr>
<tr>
<td>Level 0</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Level 1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Level 2</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

The AVOID system is tested under varying conditions which is experienced by a driver. This system is tested based on different lighting conditions while driving. The appearance factors are also considered while testing eye blink detection and yawning detection module. In facial appearance, person having beard or driver wearing glasses while driving this scenarios are considered as factors affecting system performance.

Smartphone camera is used to capture driver's instantaneous feed. Hence camera quality and frame rate of the camera are also taken into consideration while performing system testing along with external factors such as lightning and facial appearance. For each level of drowsiness that is level 0, level 1, level 2; five test cases are considered.

Table 3. Eye blink detection module tested for facial appearance factor where in driver is wearing glasses

<table>
<thead>
<tr>
<th>Lighting conditions</th>
<th>Daytime</th>
<th>Night light</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Result</td>
<td>Expected</td>
<td>Actual Result</td>
</tr>
<tr>
<td>Level 0</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Level 1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Level 2</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4. Yawning detection module tested for facial appearance factor where in driver is a male

<table>
<thead>
<tr>
<th>Lighting conditions</th>
<th>Daytime</th>
<th>Night light</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Result</td>
<td>Expected</td>
<td>Actual Result</td>
</tr>
<tr>
<td>Level 0</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Level 1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Level 2</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5. Yawning detection module tested for facial appearance factor where in driver is a female

<table>
<thead>
<tr>
<th>Lighting conditions</th>
<th>Daytime</th>
<th>Night light</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Result</td>
<td>Expected</td>
<td>Actual Result</td>
</tr>
<tr>
<td>Level 0</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Level 1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Level 2</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

In case of yawn detection, an actual yawn is not always detected properly. This may be due to the condition that the driver opens his/her mouth too wide while yawning such that the basic structure of the face is disturbed and not recognized by the facial landmark detector. This causes decrease in the accuracy of yawn detection.

Table 6. Eye blink detection module Accuracy

<table>
<thead>
<tr>
<th>Lighting conditions</th>
<th>Daytime</th>
<th>Night light</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver is not wearing glasses</td>
<td>93.33%</td>
<td>53%</td>
</tr>
<tr>
<td>Driver is wearing glasses</td>
<td>73.33%</td>
<td>40%</td>
</tr>
</tbody>
</table>
Table 7. Yawning detection module Accuracy

<table>
<thead>
<tr>
<th>Lighting conditions</th>
<th>Daytime</th>
<th>Night light</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facial appearance factor in yawning module where in driver is a male</td>
<td>86.67%</td>
<td>40%</td>
</tr>
<tr>
<td>Facial appearance factor in yawning module where in driver is a female</td>
<td>93.33%</td>
<td>53.33%</td>
</tr>
</tbody>
</table>

Thus here functionality testing is performed for each and every modules such as eye blink detection module and yawning detection module. It seems that AVOID works quite well under daylight conditions. However, looking at the above results, a drawback of AVOID becomes apparent- it’s performance is not adequate in low light conditions. The accuracy for both eye blink and yawn detection drops significantly during night time.

In this output frame, output is represented in the format of number of blinks and number yawning performed in time intervals also current state of eyes and mouth is represented as State_blink for eye state and State_yawn represents current mouth state which is classified as close mouth and closed eyes for current scenario.

6. CONCLUSION

Driver drowsiness is a pressing issue and needs to be dealt with effectively. There is a need for systems to be in place to detect drowsiness and strict regulations to deal with offenses involving driving while drowsy. AVOID has proven to be a system that can help detect drowsiness and reduce the number of accidents due to it.

AVOID has its roots in Machine Learning, with a CNN being at the heart of its implementation. AVOID can be a torchbearer for further work and that someday, there will be no more fatalities due to drowsy driving.

Our proposed system focuses on a Machine learning approach to detect driver fatigue. The driver’s video is captured in real time. The face and the eyes are segmented using a Haar cascade classifier along with the dlib library. Now here fatigue is detected based upon the changes that happen in the facial features when a user is drowsy. Whenever fatigue is detected positively by our system where CNN comes into play, AVOID throws an alarm for the user with an intention to wake him up before any fatality occurs. This makes AVOID more reliable and effective also being economical too.

7. ACKNOWLEDGMENTS

This research would not have been completed without the valuable contribution from various people with whom the authors interacted within the organization while completion of it. The authors would sincerely like to thank their mentor Prof. Rajni Pamnani. She rendered her invaluable experience and guidance with a touch of inspiration, discipline and motivation. The authors appreciate the steady support of the professors and assistants from the Department of Computer Engineering, K.J. Somaiya College of Engineering for their guidance, feedback, and encouragement.

8. REFERENCES


