## 

**AuE - 8930 Perception and Intelligence**

**Final Project: Literature Review**

**Title: Driver Distraction and Drowsiness Detection System**

**Students: Yong He, Aditya Gopinath, Huzefa Dossaji**

**Date: 04/03/2019**

## 

## **The importance of this problem:**

The safety of driving has always been a topic that people pay close attention to. There are many advanced driver assistance systems implemented to the vehicles to assist drivers to handle severe driving conditions. However, the distraction of drivers can not be handled by these advanced driver assistance systems. According to the statistic report from National Highway Traffic Safety Administration, there are 3,450 people killed by distracted driving in 2016 and 391,000 people were injured in motor vehicle crashes involving distracted drivers in 2015 [1]. On the other hand, the drowsy driving killed 795 people in 2017 and there are 90,000 motor vehicle crashes involving drowsy driving in 2015 [2]. Based on these data, the detection of distraction of drivers becomes extremely important. There are three aspects relating to the detection of distraction of drivers which are drowsy driving, head pose estimation and eye tracking. Here follows these three aspects.

Drowsy driving is also called fatigue driving. It is directly relating to facial and eyes expressions such as yawning and eyes closure. Based on police reports, the US National Highway Traffic Safety Administration conservatively estimated that a total of 100,000 vehicle crashes each year are the direct result of driver drowsiness. These crashes resulted in approximately 1,550 deaths, 71,000 injuries and $12.5 billion in monetary losses [3]. Drowsy driving can often be detected by the position of the drivers eyelids.

Pose estimation for heads refers to its relative position and orientation with respect to the camera. Angle and pose information of heads during driving can further improve the performance of non-distracted driving. When designing smart vehicles, head pose information can estimate the drivers gaze and level of detection of the road at that current time. This is useful in studying habits such as checking your rear-view mirrors when changing lanes.

Finally, eye tracking is used in gaze pose estimation. It is useful in determining medical afflictions like during a stroke, drowsiness or distraction that involves rapid eye movement. Eye tracking in conjecture with head pose estimation and orientation can depict overall driver concentration levels. This is necessary as it has the potential to eliminate to a large extent the accidents caused by driver negligence.

## **The challenges of this problem:**

For the detection of drowsiness of drivers, there are many conditions that make detection process become difficult. For example, when the lighting conditions are terrible, when the driver does not face towards front or when the driver wears accessories such as hats and glasses, it is hard to capture facial features or eye characteristics of the driver if the behavior-based approach is implemented. Besides, it is hard for these devices to distinguish between spontaneous expressions due to fatigue of drivers and intended expressions and movements such as eyelid closure due to sleepiness or intended expression while talking or singing. (There are two other approaches for detection of drowsy driving. The first one is physiological status-based approach which utilizes physiological activities of the human body such as brainwave, heart rate and muscle signal to detect the drowsiness. The second one is vehicle behavior-based approach which measures the vehicle behaviors such as stir movement, driving lane change and brake and gas pedal behavior to predict the drowsiness of drivers. In this project, the visual-based approach with deep learning is what is mainly dived deep so these two approaches are not be discussed in this paper).

One of the most obvious challenges that comes up when trying to estimate pose and angle position for the head while driving is not having a camera that gives enough information about the image. This is referring to the absence of depth information in a typical RGB camera. Since most people do not have access to stereo cameras, and the associated processing time required to estimate head pose position is very computationally expensive, creating algorithms that use just RGB images serves as a compromise in both speed and accuracy [4]. “While current head pose estimation (HPE) algorithms are suitable for many applications under controlled conditions, the performance drops in driving environments where images commonly have varying illumination, occlusions, and extreme head rotations” [5]. This also extends to variations in personal appearances such as skin color, hair and shifting lighting conditions causing shadows which could complicate calculations in the head pose estimation system. A couple of studies have shown that the current algorithms are extremely susceptible to failure when they encounter drivers with glasses. Another issue to just estimating head pose to detect distracted driving is that sometimes the pose or angle of the head does not necessarily correlate to a specific state of focus. Therefore “Head pose estimation is intrinsically linked to visual gaze estimation, the ability to characterize the direction in which an eye is focused. Alone, head pose provides a coarse indication of gaze, and one that can be estimated in situations where the eyes of a person are not visible (such as low-resolution imagery, or in the presence of eye-occluding objects like sunglasses). The addition of eye-gaze information would provide a better indication of gaze direction. but eye-gaze is only meaningful in conjunction with head pose information, as a person’s prediction of gaze comes from a combination of both head pose and eye direction” [6]. Another challenge with detecting distracted driving in general is the constant alerts from the vehicle on false positives. Being alerted to every instance of potential distracted driving is very bothersome and so a solution there would also be needed.

In terms of eye tracking, accurate measuring depends on light intensity and choosing the right algorithm. Accurate eye tracking also requires low noise, high resolution cameras which are expensive. Low frame rate cameras affect the performance of the algorithms and can give faulty results. Contrast in cases of bad lighting needs to be enhanced and the image needs to be denoised for the edge detection algorithms to detect the borders of the eye correctly. Also, sometimes the driver may intentionally move his head or eyes while interacting with co passengers and may not necessarily be distracted which brings rise to another challenge. The problem with detecting distracted driving in general is the constant alerts from the vehicle on false positives. Being alerted to every instance of potential distracted driving is very bothersome and so a solution there would also be needed.

**Existing solutions for this problem:**

Modern approaches relating to machine learning and deep learning to detect the driver drowsiness abandon the limitations of conventional algorithms. On the other hand, the traditional machine learning methods require us to identify the features manually which means they are unable to exploit the complex relationships between various facial or eyes characteristics in our project. Thus, deep learning algorithm has been more expropriated currently. In [7], the authors implemented a Convolutional Neural Network based representation feature learning approach to classify the driver as drowsy or non- drowsy state based on an images dataset including different physical attributes under different light conditions. In [8], the authors presented a deep architecture called Deep Drowsiness Detection(DDD) network including three deep networks which are AlexNet, VGG-FaceNet and FlowImageNet to detect driver drowsiness from input video. These three networks can help to extract robust images from various environments and backgrounds, robust images indicating different facial characteristics and also extract movement patterns such as face and head gestures.

To track head pose, the easiest and most straightforward way to estimate different positions from a static image from a RGB camera, is to compare that new head image to a bunch of labeled training examples and see if there is a similar one. “Systems have been proposed that compare these views using normalized cross-correlation, mean squared error, differences in gradient direction, elastic graph matching, and distance between subspace projections” [6]. These approaches are appealing because of its lack of complexity and the need for only positive examples for the training data. However, the estimation strategy that seems to invoke a “winner-take-all pose estimation” is extremely susceptible to noise [6]. Also, this approach could be quite time intensive to have to search through a dataset every time step to find a match. Thus, the system would greatly benefit if it incorporated deep learning to match a specific facial expression and pose to one in its trained set. That is why another approach used is “machine learning techniques to train an array of face detectors each attuned to a specific pose direction. These detector array systems have consisted of neural networks, Adaboost cascades, view-based subspace energy detectors, and support vector classifiers in a kernel principal component analysis subspace” [6]. However due to complexity issues in training many detectors, only a few discrete poses are used to detect pose regardless of differing drivers faces.

For eye tracking, current approaches include GPF (gradient projection function) algorithm, CDF (cumulative distribution function algorithm) and the EA (edge analysis) algorithms. For low detection errors, GPF is most efficient but it fails in cases where image contrast is low. Projection functions that are generally used are the integral projection function and the variance projection function. A general projection function is also used that combines the two along with an appropriate weight. The derivative of this function compared to an appropriate threshold can give the gradient change from the outer edge of the eye to the pupil. This is then averaged from the bounding coordinates to determine the eye center. CDF works well all round as its not very dependent on image contrast as compared to EA or GPF. Edge detection will utilize gaussian blurring and then application of canny or Sobel filter to identify the edges [9].

**Proposed solutions for this problem:**

For our solution we propose to use the DeepGaze library from mpatacchiola on Github to implement head pose, gaze estimation and possibly driver drowsiness as well. This library is based on TensorFlow and OpenCV and is a machine learning computer vision library for human-computer interaction that makes use of Convolutional Neural Networks and other types of algorithms for all types of classifications. For example, when we are trying to estimate head pose, this library has the capability of not only classifying images using CNN but also Perspective-n-Point algorithm with dlib face detectors, and Perspective-n-point in OpenCV [10][11].

Referring to how the head pose estimation in DeepGaze packages work, “One common way to estimate head pose is to use the Perspective-n-Point (PnP) [12] method to solve the correspondence between the 2D landmarks of faces and a 3D generic head model. A rotation matrix is thus obtained and can be transformed to angle representations” [4]. The PnP problem just refers to “estimating the pose of a calibrated camera given a set of n 3D points in the world and their corresponding 2D projections in the image” [13].

For driver drowsiness detection, there is a real-time drowsiness detection approach based on computer vision [14]. It detects the degree of eyes closure based on the Eye Aspect Ratio. The algorithm is representing each eye through 6 (x,y) coordinates which are shown as followed:

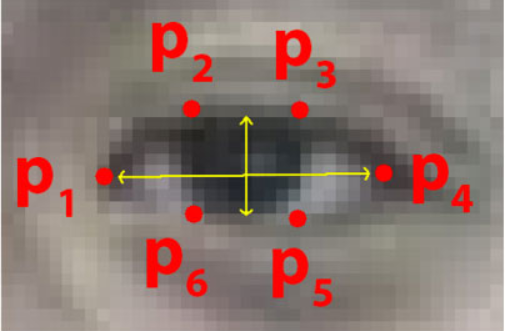


Figure 1: Eye coordinates [14]

The relationship between these 6 points is represented by the Eye Aspect Ratio:

This method checks 20 consecutive images from input video and if the EARs are less than 0.25, alert is generated to represent the drowsy driving. This approach is based on Python including dependencies which are openCv, Dlib, scipy and imutils. Although this approach will be explored it has many limitations. If needed to satisfy many different environments and conditions to detect driver drowsiness, the DeepGaze library will again come into play as we will either use the skin and color detection packages to identify when a eyelid is closing, or create another model which uses a Convolutional Neural Network to train to detect when a person seems drowsy based on the position of their eyelids. If a real time approach is taken as explained above, then it will rely heavily on the eye gaze detection as well where a gradient particle filter algorithm is used to calculate the gradient changes of the cornea to the pupil and get the localized coordinates of the eye. The points are then averaged to find the center point of the pupil [9]. Using a specified condition or detection parameters in terms of x and y coordinates for the drivers’ eye as well as pupil dilation detection, we can determine or predict driver drowsiness. However a better approach would be to use a CNN for detection of eyelids and predicting whether the driver is aware of his environment or not [15].

In terms of hardware and computing requirements, the proposition is to use a camera feed with a frame rate of 20 FPS or higher for image processing algorithms to perform at required efficiency. For this purpose a raspberry pi camera is being considered as it will allow for seamless data collection and processing once the model is built on the PC. If this hardware fails we will resort to using webcam data from the PC itself. The contrast of the feed from the cameras will be improved for enhanced detection using histogram equalization or similar related method.

As a future addition, this concept can be extended to alert the driver in case of a hazard approaching the vehicle or vice versa but only when distracted driving is detected. This will eliminate bothersome alerts for distracted driving when no hazards or danger is imminent. This method would require sensor fusion of the detection system with the radar or distance sensors to activate a sensory warning. For demonstration, this can be shown with the help of a microcontroller, ultrasonic sensor and a buzzer/LED.

**References:**

[1] Andrew.currin.ctr@dot.gov. “U Drive. U Text. U Pay.” NHTSA, 20 Dec. 2018, [www.nhtsa.gov/risky-driving/distracted-driving](http://www.nhtsa.gov/risky-driving/distracted-driving).

[2] Andrew.currin.ctr@dot.gov. “Drowsy Driving.” NHTSA, 18 Dec. 2018, [www.nhtsa.gov/risky-driving/drowsy-driving](http://www.nhtsa.gov/risky-driving/drowsy-driving).

[3] “The Zebra's 2018 Distracted Driving Report.” Compare Car Insurance Quotes: Fast, Free, Simple, [www.thezebra.com/distracted-driving-report-2018/](http://www.thezebra.com/distracted-driving-report-2018/).

[4] Hsu, Heng-Wei, et al. "QuatNet: Quaternion-Based Head Pose Estimation with Multiregression Loss." IEEE Transactions on Multimedia 21.4 (2019): 1035-46. CrossRef. Web.

[5] Jha, Sumit, and Carlos Busso. "Challenges in Head Pose Estimation of Drivers in Naturalistic Recordings using Existing Tools".IEEE , Oct 2017. 1-6. Print.

[6] Murphy-Chutorian, E., A. Doshi, and M. M. Trivedi. "Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation".IEEE , Sep 2007.

[7] K. Dwivedi, K. Biswaranjan and A. Sethi, "Drowsy driver detection using representation learning," 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, 2014, pp. 995-999.

[8] Park S., Pan F., Kang S., Yoo C.D. (2017) Driver Drowsiness Detection System Based on Feature Representation Learning Using Various Deep Networks. In: Chen CS., Lu J., Ma KK. (eds) Computer Vision – ACCV 2016 Workshops. ACCV 2016. Lecture Notes in Computer Science, vol 10118. Springer, Cham

[9] Ciesla, Michal, and Przemyslaw Koziol. "Eye Pupil Location using Webcam." (2012)Web.

[10] Mpatacchiola. “Mpatacchiola/Deepgaze.” GitHub, 26 Mar. 2019,

Mpatacchiola. “Mpatacchiola/Deepgaze.” GitHub, 26 Mar. 2019, github.com/mpatacchiola/deepgaze.

[11] Patacchiola, Massimiliano, and Angelo Cangelosi. "Head Pose Estimation in the Wild using Convolutional Neural Networks and Adaptive Gradient Methods." Pattern Recognition 71 (2017): 132-43. CrossRef. Web.

[12] Yinguobing. “Yinguobing/Head-Pose-Estimation.” GitHub, 6 Mar. 2019, github.com/yinguobing/head-pose-estimation.

[13]“Perspective-n-Point.” Wikipedia, Wikimedia Foundation, 6 Mar. 2019, en.wikipedia.org/wiki/Perspective-n-Point.

[14] akshaybahadur21. “akshaybahadur21/Drowsiness\_Detection.” GitHub, 25 Sept. 2018, github.com/akshaybahadur21/Drowsiness\_Detection.

[15] Balya, David, and Tamás Roska. "Face and Eye Detection by CNN Algorithms." Journal of VLSI signal processing systems for signal, image and video technology 23.2 (1999): 497-511. Web.