**Detecting Drowsiness in Automobile Drivers**

Preventing Accidents by Identifying Behavior Associated with Drowsiness

Aaron Davis  
 Data Science  
CU Boulder  
 Boulder, CO, USA

Bhawneet Singh  
 Data Science  
CU Boulder  
 Boulder, CO, USA

Karan Dhir  
 Data Science  
CU Boulder  
 Boulder, CO, USA

Poonam Parag Thakur  
 Data Science  
CU Boulder  
 Boulder, CO, USA

**ABSTRACT**

Accidents caused at least partially by drowsy drivers are one of the major causes of preventable deaths globally. The goal of this project is to use machine learning to evaluate the alertness of an automobile driver and to provide feedback to the driver about whether or not they should be driving.

**CCS CONCEPTS**

• Applied Computing • Life and Medical Sciences   • Consumer Health

**KEYWORDS**

machine learning, convolutional neural networks, driver safety, drowsy, tired, alert, real-time, video, analysis

**ACM Reference format:**

Aaron Davis, Bhawneet Singh, Karan Dhir, and Poonam Parag Thakur. 2021. Detecting Drowsiness in AutomobileDrivers*: Preventing Accidents by Identifying Behavior Associated with Drowsiness.* *3 pages*

**1 Introduction**

According to a poll taken by the CDC in 2009-2010, over 4% of drivers admitted to having fallen asleep while driving within the past 30 days [1]. Furthermore, an estimated 16.5% of fatal accidents involve drowsy drivers [2]. Drivers falling asleep seems to be a pressing issue that needs to be addressed to mitigate further unnecessary loss of life. The goal of this project is to provide one potential solution to this pressing issue.

We want to provide a highly accurate machine learning model that can detect drowsiness in real time based on a video feed, with all data processed on-device, so that there are no potential privacy concerns involved in using the machine learning model. We then intend to use the classifications that the model outputs over time to alert the driver that they should not be driving, currently.

In this project, we will use machine learning image classification techniques (e.g. convolutional neural networks) to classify whether or not a driver is tired based on a real-time video feed of the driver’s facial expressions and head positions over time.

Our classifier model needs to be computationally conservative, so that it can be run on a small, cheap computer, like a Jetson Nano or Raspberry Pi 3/4. Future work may involve designing a casing to contain such a computer running our model, with a built-in alarm to wake drowsy drivers, and even potentially a built-in motion sensor to detect vehicle weaving, which could serve as another indicator of drowsiness.

**2 Related Work**

2.1 Drowsiness Estimation Using a Heart Rate Sensor

Lakshmi Priya et al. [3] monitor a heart rate sensor to determine whether or not to turn on the camera and check for signs of drowsiness. They determine drowsiness using the eye aspect ratio (EAR) which is a measure of how open or closed the eyes of the driver are.

This study is different from our planned work in a couple of big ways. First, this related work uses a heart rate sensor. Our proposed work does not. We are trying to design something that requires as little pre-driving setup as is possible. Second, this related work stores data about the user on the cloud. Again, we have no plans to go this route for privacy reasons.

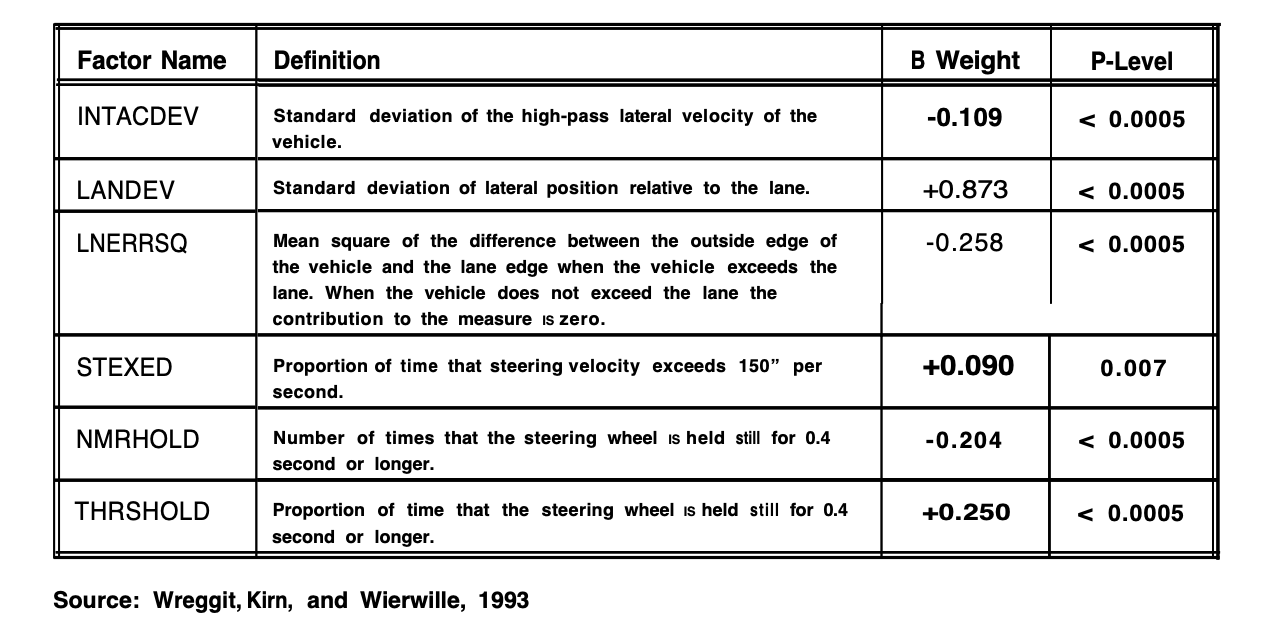
2.2 Drowsiness Estimation using Respiratory Monitoring

Guede-Fernández et al. [4] attempted to detect drowsiness using respiratory monitoring. The method used to measure breathing over time was respiratory inductive plethysmography (RIP).

This setup requires the monitored driver to wear a respiration-monitoring harness. While this study yielded decent results (Cohen’s Kappa agreement score of 0.75± 0.19), the pre-driving setup required makes this drowsiness detection method so inconvenient that we do not believe it can be successfully scaled to be used by the general population.

2.3 Drowsiness Estimation using Vehicle Metrics

Knipling & Wierwille [6] evaluated the ability of several vehicle motion metrics to determine driver drowsiness levels. They found that the standard deviation of the lateral position of the vehicle relative to the lane is the single strongest predictor of driver drowsiness.



Once our planned image classification algorithm is completed and tested, we may add an accelerometer to our setup to measure the change in lateral position over time. We could potentially use this information in our study as a second indicator of driver drowsiness.

2.4 Review of Available Methods

Sahayadhas et al. [5] reviews known methods for detecting drowsy drivers. The first proposed method of detection is through driving style or vehicle movement data. This includes information about how the is steering or accelerating.

The second proposed method of detection is via the driver’s behavior. This could include detecting things like eyelids drooping, yawning, or head slowly dropping due to overly relaxed muscles.

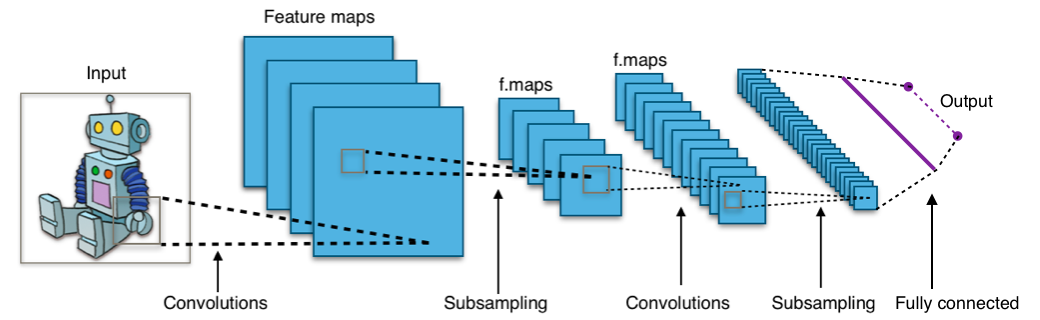
The third proposed method of detecting drowsy drivers is through physiological changes, like changes in heart rate or eye movement speed (rapid or slow eye movement).

**3 Original Proposed Work**

In real-time, we will acquire and process video of the driver’s face and head positions. Then we will split the video into frames and use a machine learning model to classify each image as drowsy or not drowsy.

We will do this by identifying key facial characteristics using image segmentation and then measuring key features like eye aspect ratio (i.e. how open or closed the driver’s eyes are). Other interesting metrics like the amount of time per blink or the resting position of the head may also be evaluated.

Currently, we are planning on using convolutional neural networks for image segmentation and drowsiness classification.



By Aphex34 - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=45679374>

Convolutional neural networks (CNNs) use layers of learned “slides” to essentially condense small parts of images into a summarizing set of numbers. [7] This process is repeated over and over until the result is fed into a fully-connected artificial neural network to output the image classification or regression (depending on the use case). This process allows the CNN to learn to identify complex features in images, without using the incredibly large number of neurons that a fully-connected artificial neural network would need to have to solve the same classification/regression problem.

The dataset we will be training our CNN model on contains 2467 images for training, split evenly between eyes open, eyes closed, yawning, and not yawning. Further, we also have access to a testing dataset that contains 433 images, also evenly split between the four classes represented in the training dataset. While our train and test datasets are split evenly between these datasets, in real life the split between drowsy and non-drowsy drivers will not be balanced - there will be more non-drowsy drivers in actual use than there will be drowsy drivers. This presents some potentially interesting challenges for evaluating our model.

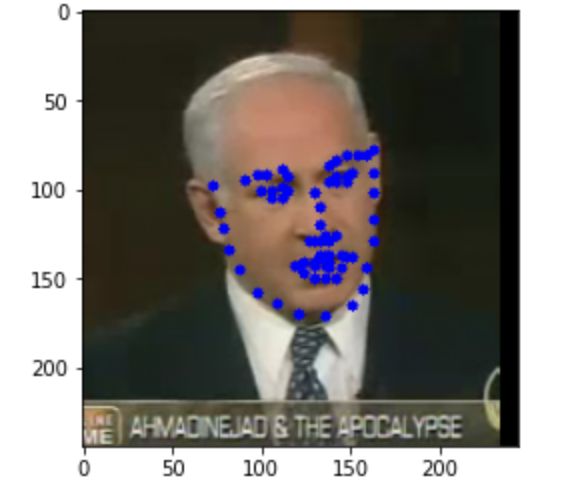
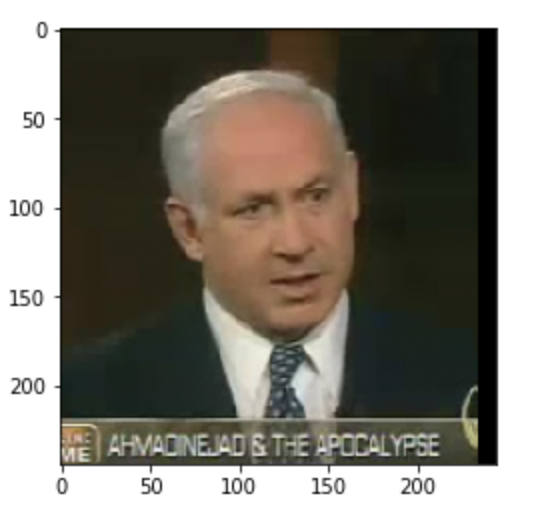
**4 Final Work**

4.1 Difficulties with original approach

We worked to implement an eye and face detector for our project using haar cascades and quickly discovered that that approach was highly sensitive to lighting, head orientation, and the presence of eyeglasses and face masks. We have also worked to implement a convolutional neural network trained on the 2500 images mentioned in the previous section. This model did not generalize well, and we believe that was at least partially a result of low variability in the training data.

We have modified our original approach so that our model will be more resilient as the scene lighting changes and as the driver’s head orientation changes. Our original approach attempted to evaluate drowsiness directly from images of the driver. The problem here is that the model we trained did not generalize well to real-world data, even though it achieved 95% accuracy on a validation portion of the training dataset used.

Obviously, our model needs to be able to generalize well if it is going to be used in vehicles with varying interiors and with drivers with different facial structures. With generalization as the end goal, we redesigned our approach to identify facial features relevant to detecting drowsiness (e.g. how open are the driver’s eyes). Our new approach is to identify the coordinates of facial keypoints using a CNN trained on labeled facial keypoint data from YouTube provided by Wolf [8] and Bulat [9]. See the following images for an example of one of the labeled images from this dataset.



This dataset includes over 17 gigabytes of data from YouTube videos. We downsampled this image data to include one frame from each video, along with several augmented frames per original frame. The augmented frames were rotated by a random angle between -45 and 45 degrees and then blurred a random amount using a gaussian blurring function.

The goal of this downsampling and augmentation is to help minimize dataset size while maintaining (or even increasing) the total variance of the dataset. The goal of increasing variance is to help the model generalize to new, never-before-seen situations.

4.2 Challenges to Current Approach

Building a model that can reliably identify facial keypoints in spite of varying head positions and lighting conditions is the real challenge of this project. Identifying drowsiness and alerting a driver is a trivial problem in comparison.

The main issue with our current approach (using a CNN to explicitly identify facial keypoints before attempting to identify drowsiness) again stems from our trained model not generalizing well beyond the training dataset. In spite of this shared weakness, our new approach is still superior to our previous approach for one big reason; improved model interpretability. Labeling and viewing key facial points before classifying drowsiness will help us determine why the model is classifying someone as drowsy by introducing an easily interpretable intermediate step.

Kartynnik, et al. [10] identify facial key points by 1) using a separate fast face detection model to find the faces in an image, then 2) rotating the cropped image so that the eyes are horizontal, and finally 3) training a neural network to map facial keypoints onto the cropped image. The image transformation from step 2 is then reversed so that the keypoints can be placed back onto the uncropped, unskewed image.

Our approach is to work to replicate the results of this study without reusing their methods. As a result, we skipped the step where Kartynnik cropped out the faces of interest before attempting to identity facial keypoints. Instead, we went straight to increasing variance in our training images. Because we skipped the cropping step, our model had to learn to filter out more unnecessary information around the regions of interest in the image.

In a sense, we handicapped ourselves by attempting to increase training image variance as the first step of the project, rather than first cropping out the region of interest, and then following that up by increasing variance within the regions of interest through image augmentation.

4.3 Proposed Solutions to Current Challenges

We are tackling the issues with model generalization through improved image augmentation, better batch image loading, and by using dropout regularization in both the convolutional and dense layers of the network we are training. We are currently (as I write this) training a new facial keypoint detection model using this improved approach. Though the model has not yet been trying for very long, the results so far seem promising, with a validation train data mean squared error consistently less than the train data mean squared error.

Dropout regularization helps ensure that any particular network output does not rely too heavily on one particular path through the neural network. This effect is achieved by constantly modifying the connections in the network during training.

4.4 Results of Current Work

After several hundred hours of training and testing varying CNN architectures (many of which involved transferring learning on MobileNet V2 to help gain maximum computational speed), our facial keypoint detection results were nowhere even close to matching the work of Google’s researchers [10].

This performance difference can be attributed to two primary factors. First, they cropped out the faces in the image before identifiying keypoints. By doing this, the total amount of noise that the model had to filter out was reduced significantly. Second, Google’s team of researchers used 3D models of heads to achieve better data variation and labeling.

In the end, we did get our drowsiness detection system to work fairly well, but only while using the facial keypoint detection model that was pretrained by Google in the MediaPipe library. Given that developing the keypoint detection model is the most difficult part of the project, in the first place, identifying drowsiness after having the model pretrained is a trivial programming program. The hope is that the end result will still be able to save the lives of drowsy drivers, even though we were not able to implement the facial keypoint detection model from scratch.

4.5 Plans for Future Improvement

The main plan to improve our facial keypoint detection is to drastically change the types of network architectures used during this project. The main architecture of interest that we haven’t yet tried is the feature pyramid detectors (FPNs).

Another key thing to try would be to increase the number of labeled keypoints surrounding more important facial features (like the eyes, for example). The reasoning behind this is that it is more important that the model correctly identifies the locations of the eyes than, say, the chin. Adding more keypoints around the eyes will have a similar effect to weighting the loss function so that the error around the eyes is considered more important than the error surrounding other facial features.

**5 Original Evaluation Proposition**

Our CNN model will be tested on a dataset of 430 images that the model was not trained on. The evaluation metric used will be a precision-recall curve, because this evaluation metric is well-suited for imbalanced data, and the real-life use of our project absolutely represents an imbalanced dataset.

Most of the time, drivers will not be falling asleep. But even though sleepy drivers are infrequent, the damage caused by sleepy drivers is significant. As a result, it is crucial that we successfully classify sleepy drivers even though most drivers are sleepy very infrequently.

**6 Updated Evaluation Proposition**

6.1 Evaluating Facial Keypoint Model

Our facial keypoint model will first be evaluated on a validation set from the 17 gigabyte training set, and then the model will be validated in real time by using the model on a video stream from our laptop webcams and ensuring that the output matches well with the location of our facial keypoints. Our evaluation metric for this portion of the project is mean squared error (MSE) because we are treating this part of the project as a regression task.

6.2 Evaluating Drowsiness Detection

Once our facial keypoint model has been evaluated and validated, we will detect drowsiness by taking a moving average of the eye aspect ratio across frames, and alert the driver once their eyes have been closed for longer than a certain threshold. We will determine that threshold via testing once our facial keypoint model has been evaluated successfully.

**7 Milestones**

See the table below for information about our milestones for this project.

|  |  |  |
| --- | --- | --- |
| **Date** | **Milestone** | **Status** |
| Sep 29, 2021 | Proposal | Complete |
| Oct 1, 2021 | Presentation | Complete |
| Oct 15, 2021 | Training Set Augmentation | Complete |
| Oct 21, 2021 | Image Segmentation | Complete |
| Oct 25, 2021 | Model Training  (First Iteration) | Complete |
| Nov 1, 2021 | Project Checkpoint | Complete |
| Nov 8, 2021 | Model Optimization | Complete |
| Dec 6, 2021 | Model Testing and Deployment | Complete |

**REFERENCES**

[1] Wheaton, A., Shults, R., Chapman, D., Ford, E., & Croft, J. (2014, July 4). Drowsy Driving and Risk Behaviors — 10 States and Puerto Rico, 2011–2012. www.cdc.gov. https://www.cdc.gov/mmwr/preview/mmwrhtml/mm6326a1.htm

[2] Tefft B. C. (2012). Prevalence of motor vehicle crashes involving drowsy drivers, United States, 1999-2008. Accident; analysis and prevention, 45, 180–186. https://doi.org/10.1016/j.aap.2011.05.028DOI:https://doi.org/10.1145/567752.567774

[3] Lakshmi Priya, B., Prithviraj, M., Baraniraj, C., & Duraikannu, P. (2018). Accident Prevention System using Driver Drowsiness Detection. International Journal of Innovative Science and Research Technology, 3(7). <https://ijisrt.com/wp-content/uploads/2018/07/Accident-Prevention-System-using-Driver-Drowsiness-Detection-1.pdf>

[4] F. Guede-Fernández, M. Fernández-Chimeno, J. Ramos-Castro and M. A. García-González, "Driver Drowsiness Detection Based on Respiratory Signal Analysis," in IEEE Access, vol. 7, pp. 81826-81838, 2019, doi: 10.1109/ACCESS.2019.2924481.

[5] ‌Sahayadhas, A., Sundaraj, K., & Murugappan, M. (2012). Detecting Driver Drowsiness Based on Sensors: A Review. Sensors, 12(12), 16937–16953. https://doi.org/10.3390/s121216937

[6] ‌Knipling, R., & Wierwille, W. (1994). Vehicle-based drowsy driver detection : current status and future prospects.

[7] ‌O'Shea, K., & Nash, R. (2015, November 26). *An introduction to convolutional neural networks*. arXiv.org. Retrieved September 29, 2021, from https://arxiv.org/abs/1511.08458v1.

[8] Wolf, L., Hassner, T., & Maoz, I. (2011). *Face Recognition in Unconstrained Videos with Matched Background Similarity.*IEEE Xplore temporarily unavailable. Retrieved November 12, 2021, from https://ieeexplore.ieee.org/document/5995566.

[9] Bulat, A., & Tzimiropoulos, G. (2017, September 7). *How far are we from solving the 2D & 3D face alignment problem?*arXiv.org. Retrieved November 12, 2021, from https://arxiv.org/abs/1703.07332.

[10] Kartynnik, Yury & Ablavatski, Artsiom & Grishchenko, Ivan & Grundmann, Matthias. (2019). Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs.