2018 IEEE International Conference on Systems, Man, and Cybernetics

Driver Distraction Recognition Based On Smartphone Sensor Data

Jie Xie
Electrical and Computer Engineering
Jiangnan University and University of Waterloo, Waterloo, Canada
Email: xiej8734@gmail.com

Allaa R.Hilal
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
Email: ahilal@uwaterloo.ca

Dana Kulic
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
Email: dana.kulic@uwaterloo.ca

Abstract—Driver distraction is one of the leading causes of vehicle accidents and injury. Automated systems for identifying driver distraction are of great interest for improving road safety. This study develops a smartphone sensor based driver distraction system using an ensemble learning method. After data collection, linear velocity data is first linearly interpolated. Then, 3-axial acceleration and 3-axial gyro signals are filtered for reducing noise. Next, a sliding window is applied to IMU and GPS data collected by the smartphone for feature extraction, where temporal features are calculated. Ensemble learning of four standard classifiers is used to recognize distraction events: K-Nearest Neighbor, Logistic Regression, Gaussian Naive Bayes, Random Forest. To evaluate the proposed approach, 24 drivers were recruited to participate in a user study, driving on a route consisting of suburban and highway driving. Driver cognitive distraction was induced by asking the driver questions while driving. The experimental results show that the best weighted F1-score of our proposed system is 87% with all smartphone sensor signals.

Index Terms—Driver distraction classification, Smartphone, Ensemble learning

I. Introduction

Driver distraction is used to describe driving while another activity diverts the driver's attention away from driving. Many factors can lead to distraction while driving: talking on a cell phone, using a navigation system, eating and drinking, talking to passengers in the vehicle, and so on. Distracted driving is often cited as one of the leading causes of accidents leading to injury [1]–[3].

Recent developments of In-vehicle Information Systems (IVISs) and mobile devices provide drivers more opportunities for interaction with these systems while driving. Therefore, driver distraction is more likely to happen, which can be a threat to road safety. From 1999 to 2008, distraction-related crashes in urban environments increased from 32.7% to 39.8% in the US [4]. It is becoming increasingly important to develop methods for detecting and preventing driver distraction. During the last decade, many methods have been proposed to solve the problem of driver distraction such as raising public awareness, enhancing legislation, and optimizing the design of IVISs to minimize the potential impact [5].

Distracted driving is commonly classified into three types: (1) visual distraction: taking your eyes off the road; (2) manual distraction: taking your hands off the wheel; (3) cognitive distraction: taking your mind away from driving [6]. Depending on the type of distracted driving, different methods have been developed for recognizing driving distraction [7], [8].

Koesdwiady et al. proposed an end-to-end deep learning solution for driver distraction recognition [7]. The data used for recognizing driver distraction was captured by a camera, which included the upper body of the driver, hand positions, and rear-part of the car. A pre-trained convolutional neural network VGG-19 was used for feature extraction. The highest test accuracy of 95% was achieved among 10 drivers using leave-one-driver out evaluation, and an average accuracy over 10 drivers of four classes (normal driving, distraction left, distraction right, distraction back) was 80%. Liao et al. developed a method for the detection of driver cognitive distraction at stop-controlled intersections and compared feature subsets and classification accuracy with those on a speed-limited highway [8]. A driving simulator was used for data collection. Three small video cameras, which were located on the dashboard were used for capturing eye movement information. Three types of features were used for the detection: scenario-based features, sensor-based features and eye movement features. The performance for stop-controlled intersections and a speed-limited highway were 95.8±4.4% and 93.7±5.0%, differentiating non-distracted and distracted classes.

While these two studies achieved very promising performance, both studies included video data for driver distraction recognition, which might limit usability while driving at night, as well as limit uptake due to privacy concerns. Unlike video data, IMU and GPS data can be easily captured from the driver's mobile phone, and their performance is not affected by illumination. Our objective in this work is to estimate driver distraction from the driver's mobile device, using only motion data collected by the IMU and GPS sensors.

Compared to normal driving, distracted driving is often characterized by some distinct patterns such as (1) vehicle shifting, (2) erratic braking, (3) speeds which do not coincide with the current traffic flow [9]. Those patterns

2577-1655/18/$31.00 ©2018 IEEE
DOI 10.1109/SMC.2018.00144

suggest the possibility of using IMU and GPS data for driver distraction recognition. In this study, we develop a recognition system for driver cognitive distraction using only smartphone sensor data. Unlike previous studies [7], [8], our proposed system can be implemented only using a smartphone, and does not use the phone's video/camera data, so that the phone can be placed anywhere in the vehicle. Specifically, after data collection, interpolation and noise reduction are first used to help the subsequent analysis. Then, a sliding window is applied to the motion signals for temporal feature extraction including linear velocity, 3-axial acceleration, and 3-axial gyro. Next, we calculate signal features including mean, variance, median, range, interquartile range, kurtosis, skewness, root-mean-square, percentiles, slope, mean absolute deviation, zero-crossing rate, curvature, and histogram of skewness. For the recognition, an ensemble of four standard classifiers is used: K-Nearest Neighbour (K-NN), Logistic Regression (LR), Gaussian Naive Bayes (NB), Random Forest (RF). The weighted F1-score of our best performing model is 87%, when the window size and overlap are 80s and 50%.

II. Methods

Our driver distraction recognition system consists of four steps: data collection and preprocessing, feature extraction, classifier training, and recognition (Fig. 1). Detailed information about each step is provided in the following sections.

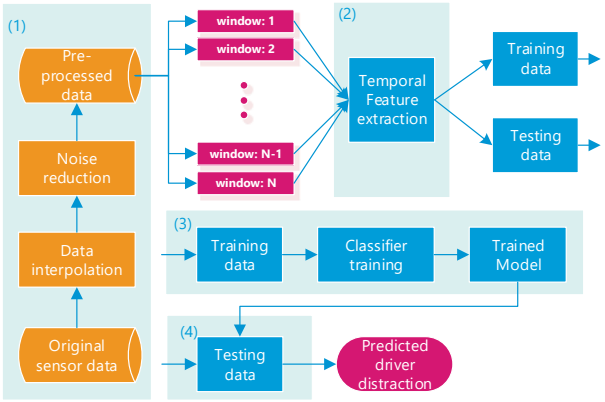


Fig. 1: Flowchart of our driver distraction recognition system: (1) data collection, (2) feature extraction, (3) model training, (4) recognition.

A. Data collection

Our data is collected in Waterloo, Ontario. 24 participants (8 females and 16 males) were recruited to participate in the study. Their age ranged from 21 to 52, and the average driving experience ranged from 1 year to 26 years. An iPhone with the SensorPlay application was placed on the floor of the front seat for collecting sensor data. The IMU data was sampled at 100 Hz. The linear velocity data (from the GPS) was available at 1 Hz, which were re-sampled at 100 Hz and linearly interpolated to maintain a constant measurement interval and consistency of sample rates during training [10]. Audio of the vehicle interior was recorded with a microphone placed on the bottom of the rearview mirror. The vehicle was also equipped with an outside facing camera, GARMIN VIRB, for driving maneuver inspection and labeling.

All participants drove the same car, a Fiat 500 (see Fig. 2) in three road conditions: parking lot, highway and city road. The participants were accompanied by two researchers who were vehicle passengers. Driver cognitive distraction was induced by asking the participants questions while driving. All data was manually labeled as normal driving and distracted driving based on the collected audio files. When the driver and the passenger are talking to each other, the conversation was labeled as distracted. intermittent driver speech with no passenger response that is shorter than three seconds were labeled as undistracted driving. Since we apply a sliding window to each signal for feature extraction, the ground truth of each window is provided based on the original labeled files using Algorithm 1. Besides distraction labels, maneuvers including stop, left turn, right turn, left lane change, right lane change, and lane keeping are manually labeled for all trips based on the outside camera.

Fig. 2: Experimental Car (Intelligent Mechatronic Systems, Inc. (IMS), Waterloo based).

The sensor information used for calculating temporal features is shown in Table I. In addition, we include the magnitude of the linear acceleration and angular velocity for feature extraction, which are calculated as follows:

$$s = \sqrt{s_x(t)^2 + s_y(t)^2 + s_z(t)^2} \quad (1)$$

Here s represents the linear acceleration or angular velocity magnitude.

B. Feature extraction

Before feature extraction, noise reduction is applied using the following method: signals are first filtered by a moving median (outlier removal from signal) and then a

TABLE I: Sensor information and Notations.

Sensor Information	Notations	Sensor Information	Notations
Linear velocity (km/h)	v(t)	Gyro in X (rad/s)	$g_x(t)$
Acceleration in X (m/s^2)	$a_x(t)$	Gyro in Y (rad/s)	$g_y(t)$
Acceleration in Y (m/s^2)	$a_y(t)$	Gyro in Z (rad/s)	$g_z(t)$
Acceleration in Z (m/s^2)	$a_z(t)$		

Algorithm 1: Label generation for our dataset

Input : Start and stop talking timestamps (t_s and t_e) of each windowed signal. θ is empirically set at 0.5. $t_{win}(k)$ is the duration of the sliding window k.
Output: Label of each data window k: $L(k)$.
begin
  Step 1: Calculate talking time: $t_d = t_e$ - t_s
  Step 2:
  if $t_d \geq t_{win} \times \theta$ then
    $L(k) = 1$
  else
    $L(k) = 0$

moving mean (smoothing filtered signal) filter. Both the order of median filter and the smoothing window size are set to five samples.

TABLE II: Functions for calculating temporal features.

#	Function	Description
1	mean	mean of a signal
2	min	minimum value of a signal
3	max	maximum value of a signal
4	var	variance of a signal
5	median	median of a signal
6	range	difference between the max and min
7	min-m	difference between the mean and min
8	max-m	difference between the max and mean
9	iqr	interquartile range
10	prctile	the 25^{th} and 75^{th} percentiles
11	sk	skewness of a signal
12	kur	kurtosis of a signal
13	slope	ratio between the range and the location of maximum and minimum
14	mad	mean absolute deviation of a signal
15	curv	curvature of both half signals and the whole signal
16	sk-hist	skewness of the histogram of a signal
17	jerk	standard deviation of the derivative of an acceleration signal

For each sliding window, we use 19 temporal features for characterization, where the 17 feature functions are shown in Table II. All those features are calculated over the whole window and have been used in [11]. In addition, the zero-crossing rate and short time energy are included in our feature set, which are calculated based on the frame.

(18) Zero-crossing rate (zcr): zero-crossing rate denotes the rate of signal change along a signal. When adjacent signals have different signs, a zero-crossing occurs. The mathematical expression of zcr is shown as follows.

$$zcr = \frac{1}{2}\sum_{n=0}^{L-1}[sgn(x(n)) - sgn(x(n+1))] \quad (2)$$

where $x(n)$ is the framed signal, L is the length of the frame.

(19) Short time energy (ste): short time energy is defined as the sum of intensity of signal.

$$ste = \frac{1}{L}\sum_{n=0}^{L-1} x(n)^2 \quad (3)$$

For zero-crossing rate and short time energy, the frame size and overlap are 20% and 40% of the duration of a sliding window. Then, mean, median and variance are calculated over the frames within a sliding window as features.

After feature extraction, all features are concatenated together to form a feature vector of dimension 273. Then, the normalization is conducted as follows:

$$v_i = \frac{v_i - \mu_i}{\sigma_i} \quad (4)$$

where μ_i and σ_i are the mean and standard deviation computed for each feature vector v_i.

C. Classification

1) Four standard classifiers: Four standard classifiers are used for the recognition: K-NN, LR, NB, and RF.

In a K-NN classifier, the distance between an input feature vector and all stored feature vectors is first calculated. Then k closest vectors are selected to determine the label of the input feature vector by majority voting [12].

LR uses the calculated logits to predict the target class, where the logits are the likelihood occurrences after being passed through logistic functions [13].

NB classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Then, Bayes theorem is used to calculate posterior probability for the classification and the class with the highest posterior probability is the outcome of prediction [13].

RF is a tree-based algorithm, which builds a specified number of classification trees without pruning. After training, it gives the output based on the mode of the classes of the individual trees [14].

2) Ensemble learning: Different recognition systems using different classifiers often make different recognition outputs for the same dataset. A combination of those classifiers often achieves a better result than any one of the participating systems [15]. By assigning weights to the results of different individual classifiers, the final output is the one with the highest total scores [16]. In this study, ensemble learning, which returns labels as argmax of the sum of predicted probabilities is used to improve

the recognition performance when compared to a single classifier.

3) Evaluation: For each classifier output, a label is assigned to each sliding window. Then, we convert sliding window labels back to sample labels for the evaluation (Fig. 3). When the sum value of each column is bigger than one, the corresponding sample is labeled as distracted driving. Compared to window-based evaluation, sample-based evaluation can provide a more accurate recognition result. The proposed system is evaluated using a weighted F1-score which is defined as follows:

$$F1\text{-}score = \sum_{i=1}^{n} 2 \cdot \frac{precision(i) \cdot recall(i)}{precision(i) + recall(i)} * r_i \qquad (5)$$

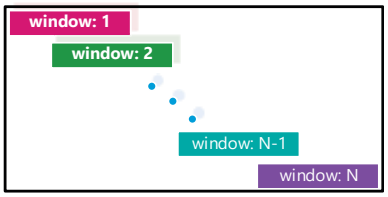


Fig. 3: Converting sliding window labels to sample labels. For each window, the label is zero (normal driving) or one (distracted driving). Here, each column of this matrix denotes one sample.

where $F1\text{-}score$ denotes the weighted F1-score, $precision$ is defined as $\frac{TP}{TP+FP}$, and $recall$ is defined as $\frac{TP}{TP+FN}$, TP is true positive, TN is true negative, FP is false positive, FN is false negative; i is the class index, r_i is the ratio between the number of samples of one class and total number of samples in all classes. For the system evaluation, we use leave one-driver-out validation to ensure that the model generalizes to new drivers.

III. Experimental results

A. Parameter tuning

For classifiers, the default settings of the scikit-learn package [17] are used for LR and NB. For K-NN and RF, the parameter settings are listed in Table IV. Grid search is used to optimize the parameters with 5 fold cross-validation.

B. Results

The weighted F1-score with different sliding window sizes, window overlaps, and labeling methods is shown in Table III. The best performance (87%) is achieved using all-labeling method, when the window size and overlap are 80s and 50%. Considering the three labeling methods, the all-labeling method achieves the best result, while half-labeling is the worst. Since our evaluation is based on samples rather than windows, half-labeling often has a gap between the window label and the sample label. Considering the three window overlaps, a window overlap of 50% achieves the best performance for both all-labeling and half-labeling. One reason might be that the same maneuver will belong to different windows using a big overlap. However, no overlap might miss some important patterns of distraction. For all window overlaps, a window size of 60s or 80s achieve the highest F1-score for different combinations of labeling and overlap. Regarding different window sizes, a small window size cannot capture sufficient information to represent the distraction pattern. Meanwhile, a big window size will reduce the size of training data and might include more than one class within one window.

The confusion matrix of the best performing model is shown in Fig. 4. 16% of distraction samples are confused with normal samples. On the other hand, only 9% of normal samples are misclassified as distraction samples. This result indicates that distraction samples are more easily misclassified then normal samples.

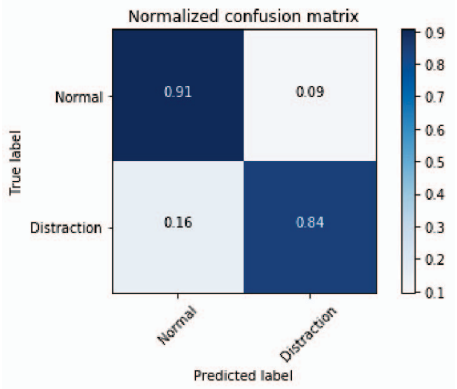


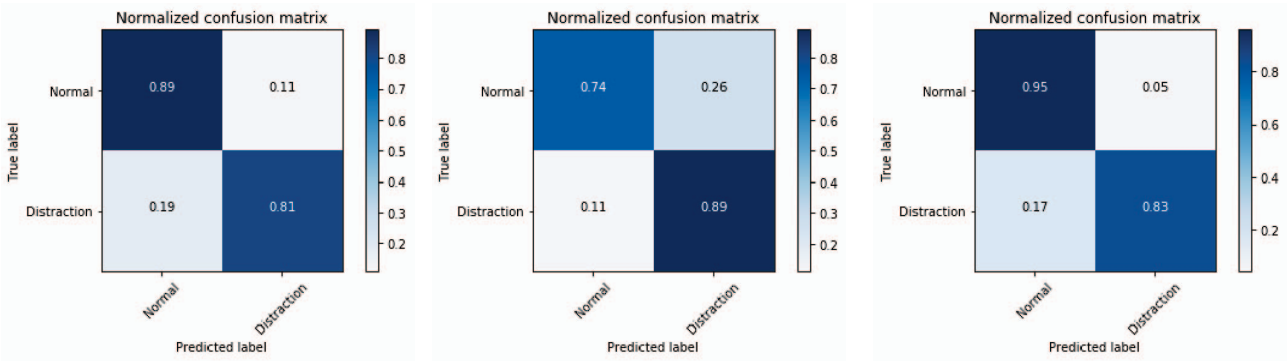
Fig. 4: Confusion matrix of the best performing model. Here, the window size and overlap are 80s and 50% overlap

We also plot the confusion matrix for three types of maneuvers (Fig. 5): stop or near-stop, lane keeping, and lateral maneuvers (turning and lane change). The F1-score for stop or near-stop, lane keeping, and lateral maneuvers are 74.3%, 83.8%, 87.3%, respectively. We can observe that the distraction classifier performance during lateral maneuvers is the best, likely because that both turning and lane changes are more likely to be affected when the driver is distracted. As expected, a lower performance is achieved for stop or near-stop. Since our model is based on the IMU and GPS data, the signals are not changing when the car is at a stop or near-stop. Therefore, it is very difficult to recognize the driver distraction.

In this study, only temporal features are used for the distraction recognition. However, spectral features have been used in previous studies for distraction recognition [18]. Here, we compare the recognition results of spectral features, temporal features and the combination of spectral and temporal features in Fig. 6. For spectral features, we calculate Mel-frequency Cepstral coefficients, spectral

TABLE III: Classification result using different sliding window sizes, window overlaps, and labeling methods. Here, the value θ is described in Algorithm 1. The best result with the same labeling method and window overlap is in bold. The value with * indicates the best performance.

Labeling method	Size / Overlap	5s	20s	40s	60s	80s	100s	120s
All label ($\theta = 1$)	No overlap	0.71	0.74	0.77	0.77	0.84	0.79	0.82
	50%	0.64	0.72	0.82	0.77	0.87*	0.80	0.80
	80%	0.74	0.64	0.67	0.83	0.78	0.71	0.74
Strong label ($\theta = 0.8$)	No overlap	0.71	0.74	0.77	0.75	0.82	0.76	0.76
	50%	0.63	0.72	0.81	0.76	0.84	0.75	0.80
	80%	0.74	0.63	0.66	0.82	0.76	0.68	0.72
Half label ($\theta = 0.5$)	No overlap	0.71	0.74	0.75	0.77	0.78	0.77	0.72
	50%	0.63	0.70	0.80	0.73	0.82	0.72	0.76
	80%	0.73	0.62	0.65	0.78	0.70	0.64	0.67



(a) Lane keeping (b) Stop or near stop (c) Lateral maneuvers

Fig. 5: Confusion matrix of the best performing model for three types of maneuvers: lane keeping, stop or near stop, and lateral maneuvers.

TABLE IV: Classifier parameter settings.

Algorithm	Parameter	Values
K-NN	K	1, 3, 5, 7, 9
RF	n_estimators	100, 300, 500
	min_samples_leaf	1, 7, 15
	max_features	'auto', 'log2', 'None'
	min_samples_split	10, 30, 50

centroid, spectral roll-off, spectral flux, and spectral entropy, whose descriptions can be found in [19]. Compared to spectral features, temporal features achieve a better performance. In addition, combining spectral features with temporal features does not help the classification result. We also compared the performance of a single classifier and ensemble learning for the four classifiers. The best weighted F1-score of RF is 84%, which is 3% smaller than with ensemble learning. This result indicates the usefulness of using ensemble learning.

We use RF to evaluate the importance of features. Since leave one-driver-out is used for the model evaluation, there are 24 classification tasks for 24 drivers. The five most important features for three randomly selected classification tasks are shown in Table V using the best performing model. We find that $g_z(t)$ and $v(t)$ are the two most important signals, with *min*, *min-m*, and *median* the

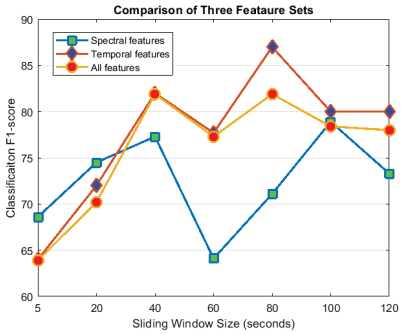


Fig. 6: F1-score of spectral features, temporal features, and all features including both spectral and temporal features. Here the window overlap is 50% and the all-labeling is used.

three most important functions. The *min* and *min-m* of $g_z(t)$ possibly capture the vehicle shifting, while the *min* , *min-m*, and *median* of $v(t)$ may detect erratic braking.

We also investigate using only $g_z(t)$ and $v(t)$ for the driver distraction recognition. The result is shown in Fig. 7. The best F1-score using only the two signals is 74.8%, which is 12.6% smaller than using all signals.

TABLE V: Feature importance of three randomly classification tasks of the best performing model. Here, FI denotes feature importance. First denotes the most importance feature, while Fifth is the least importance.

Task \ FI	First	Second	Third	Fourth	Fifth
1	min($g_z(t)$)	min-m($g_z(t)$)	median($g_z(t)$)	max($v(t)$)	curv3($v(t)$)
2	min-m($g_z(t)$)	min($g_z(t)$)	median($g_z(t)$)	sk($g_z(t)$)	max($v(t)$)
3	min-m($g_z(t)$)	min($g_z(t)$)	median($g_z(t)$)	max($v(t)$)	mad($a_x(t)$)

Compared to previous studies [7], [8], we develop a driver cognitive distraction recognition system relying only on the mobile phone, which does not require viewing the driver or knowing the driver or the outside conditions. We also do not need to perform maneuver detection, but performance is improved when the driver is performing a more difficult maneuver.

One limitation of this study is that only one cognitive distraction is tested by talking to the passenger while driving. In addition, our dataset is collected with only 24 drivers using the same vehicle.

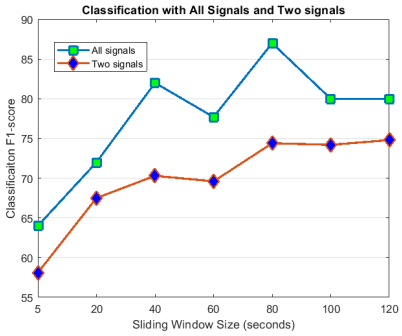


Fig. 7: Classification with all signals and two signals: angular velocity and linear velocity.

IV. Conclusion and Future work

In this paper, we develop a recognition system for driver cognitive distraction based on smartphone sensor data (IMU and GPS). After signal preprocessing, a sliding window is applied to the signals. Temporal features are then calculated for the recognition of distracted driving using an ensemble learning of four standard classifiers. The distracted driving during lateral maneuvers is more likely to be recognized than during lane keeping and stop or near-stop events. The best performing model can achieve a weighted F1-score of 87% using all signals.

Future work aims to use automatic maneuver classification algorithm for assigning maneuver labels, so that the distraction estimation can be applied only during lateral maneuvers, when observability is highest.

Acknowledgment

The authors would like to thank NSERC and Intelligent Mechatronic Systems, Inc. for their support of this research.

References

[1] S. G. Klauer, F. Guo, B. G. Simons-Morton, M. C. Ouimet, S. E. Lee, and T. A. Dingus, "Distracted driving and risk of road crashes among novice and experienced drivers," New England journal of medicine, vol. 370, no. 1, pp. 54–59, 2014.

[2] T. A. Dingus, F. Guo, S. Lee, J. F. Antin, M. Perez, M. Buchanan-King, and J. Hankey, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," Proceedings of the National Academy of Sciences, vol. 113, no. 10, pp. 2636–2641, 2016.

[3] F. Guo, S. G. Klauer, Y. Fang, J. M. Hankey, J. F. Antin, M. A. Perez, S. E. Lee, and T. A. Dingus, "The effects of age on crash risk associated with driver distraction," International journal of epidemiology, vol. 46, no. 1, pp. 258–265, 2016.

[4] F. A. Wilson and J. P. Stimpson, "Trends in fatalities from distracted driving in the united states, 1999 to 2008," American journal of public health, vol. 100, no. 11, pp. 2213–2219, 2010.

[5] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," IEEE transactions on intelligent transportation systems, vol. 17, no. 4, pp. 1108–1120, 2016.

[6] N. H. T. S. Administration. Policy statement and compiled faqs on distracted driving. http://www.nhtsa.gov.edgesuite-staging.net. Accessed: 2018-03-24.

[7] A. Koesdwiady, S. M. Bedawi, C. Ou, and F. Karray, "End-to-end deep learning for driver distraction recognition," in International Conference Image Analysis and Recognition. Springer, 2017, pp. 11–18.

[8] Y. Liao, S. E. Li, W. Wang, Y. Wang, G. Li, and B. Cheng, "Detection of driver cognitive distraction: A comparison study of stop-controlled intersection and speed-limited highway," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 6, pp. 1628–1637, 2016.

[9] J. D. Lee, K. L. Young, and M. A. Regan, "Defining driver distraction," Driver distraction: Theory, effects, and mitigation, vol. 13, no. 4, pp. 31–40, 2008.

[10] C. Woo and D. Kulić, "Manoeuvre segmentation using smartphone sensors," in Intelligent Vehicles Symposium (IV), 2016 IEEE. IEEE, 2016, pp. 572–577.

[11] J. Xie, A. R. Hilal, and D. Kulić, "Driving maneuver classification: A comparison of feature extraction methods," IEEE Sensors Journal, 2017.

[12] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," The American Statistician, vol. 46, no. 3, pp. 175–185, 1992.

[13] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in Advances in neural information processing systems, 2002, pp. 841–848.

[14] T. K. Ho, "The random subspace method for constructing decision forests," IEEE transactions on pattern analysis and machine intelligence, vol. 20, no. 8, pp. 832–844, 1998.

[15] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," Emerging artificial intelligence applications in computer engineering, vol. 160, pp. 3–24, 2007.

[16] T. G. Dietterich, "Ensemble methods in machine learning," in International workshop on multiple classifier systems. Springer, 2000, pp. 1–15.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[18] S. Jafarnejad, G. Castignani, and T. Engel, "Non-intrusive distracted driving detection based on driving sensing data," 2018.

[19] J. Xie, M. Towsey, J. Zhang, and P. Roe, "Acoustic classification of australian frogs based on enhanced features and machine learning algorithms," Applied Acoustics, vol. 113, pp. 193–201, 2016.