DrunkSelfie: Intoxication Detection from Smartphone Facial Images

Colin Willoughby, Ian Banatoski, Paul Roberts and Emmanuel Agu
Computer Science Department, Worcester Polytechnic Institute, Worcester, MA, USA
{cjwilloughby, imbanatoski, pdroberts, emmanuel}@wpi.edu

***Abstract*—Drunk Driving killed over 10,000 million people in 2015, accounting for nearly a third of all traffic-related deaths in the US. In many cases, drivers do not know they are over the limit. Passive methods to detect intoxication so that drinkers can be warned proactively, are desirable. Many young people take selfportraits (selfies) while drinking. We explore whether user intoxication levels can be inferred by image analysis and classification of selfies. We analyzed a corpus of the facial images of 53 subjects after drinking 0-3 glasses of wine, extracted features from the photographs and used machine learning to classify subjects as either sober or drunk. We found that facial lines changed significantly after consuming alcohol and that facial landmark vectors were the most predictive features. We achieved a classification accuracy of 81% using Gradient Boosted Machines for classifying subjects as either "sober" (0 or 1 glasses of wine) or "non-sober" (2 or 3 glasses of wine). Augmenting the original dataset of studio images by blurring, rotating, and altering lighting in order to capture more realistic party/bar scenarios, also improved classification accuracy. We used our intoxication classifiers to build *DrunkSelfie,* an Android application that estimates the subject's drunkenness from a selfie.**

Keywords—Intoxication detection, machine learning, self-portraits, selfies, image processing, random forest.

I. INTRODUCTION

In 2015, over 10,000 people died in accidents related to alcohol-impaired driving, accounting for 29% of all trafficrelated deaths in the US [1]. Intoxicated people have impaired cognition and decision making abilities [2], and often make decisions quickly without understanding possible consequences of their actions [3]. Moreover, drinkers frequently underestimate how drunk they are, and overestimate their ability to drive. About 40% of drinkers estimate that they are under the limit when they are actually over the limit [4].

While intoxication detection methods such as the breathalyzer are accurate, they require the user to purchase, carry and use additional hardware, which reduces compliance. Passive methods of detecting drinkers who are over the limit and notifying them in order to prevent DUIs are attractive. Smartphones, which are now owned by over 77 percent of Americans are viable platforms for intoxication detection as users would not have to purchase additional hardware.

In this paper, we investigated whether smartphone users' intoxication levels can be detected by analyzing self-portraits (called selfies). Young people frequently take selfies and 87% of US adults aged 18 to 34 years have taken a photograph of themselves and uploaded it to a social media website [7]. There are over 24 billion selfies on Google's Photo libraries [15]. In many cases, selfies are taken while drinking in social settings (Figure 1). A drinker's facial characteristics change after consuming alcohol: their face reddens and relaxes. We explored whether facial changes caused by alcohol could be distinguished from changes caused by other factors (e.g. mood, exhaustion). Intoxication detection from selfies could be the basis for an app that prevents DUIs by detecting smartphone users who are over the limit, and notifying them.

Figure 1: Faces in selfie pictures taken at bars can be analyzed to detect smartphone users' level of intoxication.

Other smartphone-based passive intoxication sensing methods have also been proposed. Gait (walk) analysis using machine learning on smartphone accelerometer and gyroscope sensor data is accurate [10,24] but many drinkers sit all night while drinking, preventing gait analysis. Detection methods that require user participation such as apps that give cognitive or memory tests [28] require the user to launch the app and take the test, reducing compliance and the chance of detection.

We analyzed a dataset of the faces of 53 subjects after drinking 3 glasses of wine, extracted facial features from the photographs and used machine learning to classify subjects as either sober or drunk. We found that facial lines changed significantly after consuming alcohol and that facial landmark vectors derived from facial landmarks were the most predictive. We achieved a classification accuracy of 81% using Gradient Boosted Machines for classifying subjects as either "sober" (0 or 1 glasses of wine) or "non-sober" (2 or 3 glasses of wine). Augmenting an original dataset of studio images by blurring, rotating, and altering lighting to better reflect real work party/bar scenarios, also improved classification accuracy. We used our intoxication classifiers to prototype *DrunkSelfie,* an Android app that estimates the user's drunkenness from a selfie.

II. BACKGROUND AND RELATED WORK

Blood Alcohol Content (BAC), which quantifies the percentage of alcohol in a person's blood, is the standard measure of intoxication [27]. BAC is a reasonable estimate of how impaired a drinker is at various intoxication levels. Breathalyzers are portable devices that measure BAC on a person's breath and are almost as accurate as blood tests. However, drinkers have to purchase breathalyzers, carry, and use them, which reduces its utility. In the US, all 50 states have a legal limit of 0.08% BAC for operating a motor vehicle [8].

Smartphone-based BAC estimation methods that require user participation: work by estimating user intoxication from user inputs. A recent study identified over 2900 alcohol-related smartphone applications [6]. Some breathalyzers have accompanying apps [5] that can be connected to smartphones via Bluetooth for easy logging, tracking and visualization of drinking trends. Smartphone BAC calculator apps estimate a

smartphone user's BAC levels from information the user inputs including their gender, weight, height, ounces of alcohol consumed and duration of drinking. Most BAC calculator apps use the Widmark formula shown below [9]:

$$\frac{Floz\,Etoh \times (8 \text{ or } 10)}{Pounds\ of\ person} - (\text{Hours since first drink} \times \text{Widmark } \beta) = BAC\frac{g\%}{ml} = BAC\frac{g}{dL} = BAC\%\,w/v$$

NOTE: Use 8 for a man, 10 for a woman.

BAC estimator apps often require information such as ounces of alcohol consumed that the user does not readily know or can wrongly estimate. Intoxicheck [28] is a mobile app that gives the user a series of reaction, judgment and memory challenges from which it estimates the user's intoxication level. Intoxication detection methods that require user participation burden the user and typically have low compliance.

Passive smartphone methods to estimate intoxication: Passive or opportunistic methods generate estimates of a smartphone users' BAC by analyzing their movements, smartphone interaction tasks or other biometric unobtrusively. Passive methods are attractive, as they introduce no additional burden on the smartphone user, which increases compliance. Gait analysis by analyzing the smartphone user's walk has recently become popular [10, 24]. Alcogait [10] analyzes smartphone accelerometer and gyroscope data using machine learning to classify the user's BAC levels, achieving an 89% accuracy using a J48 classifier. Sufoletto *et al* [24] use a Bayesian regularized multilayer perceptron neural network to classify accelerometer and gyroscope data from a user's smartphone to detect their gait changes due to intoxication. They achieve a Root Mean Square Error (RMSE) of 0.0226.

DUI app [25] is a smartphone app with user interfaces that estimates the user's BAL by measuring their motor coordination and cognition while interacting with the app.
(Pearson's correlation coefficient of 0.96 with breathalyzer measurements). DrinkSense [26] is a smartphone app that uses machine learning to infer whether an individual consumed alcohol on a given weekend by classifying data gathered from their smartphone (location, accelerometer, Wi-Fi, Bluetooth, batter, screen and app usage) with an accuracy of 76.6%.

Thermal imaging can be used to detect intoxication accurately (>90% accuracy) based on dilation of blood vessels in the nose and eyelids [12]. However, most smartphones are not equipped with thermal cameras. *Facial analysis:* work related to ours includes face recognition [13], age estimation [16], gender detection [17], and detecting tired or distracted drivers, and bad driving behavior such as lane weaving [11].

III. Our Selfie Intoxication Detection Approach

We utilized machine learning to classify the user's intoxication level into discrete ranges.

A. 3 Glasses Later Intoxicated Photographs Dataset

We utilized the 3 glasses later dataset captured by Brazilian photographer Marco Alberti [18] as our corpus of images of human faces exhibiting various degrees of intoxication. Marco captured facial photographs of 53 different people while sober and also after they have consumed one, two and three glasses of wine (sample in figure 2).

Figure 2: A subject in the *3 Glasses Later* Dataset while sober and after drinking one, two and three glasses of wine.

We note that the dataset was not gathered scientifically and may overdramatize emotions. However, the images were captured in studio lighting and were of good quality. Real world images could have faces off-center, be taken at different camera angles, and highly variable bar or party lighting. To diversify this controlled dataset, we utilized image augmentation to extend it to reflect more real world conditions including varying lighting, colors, sharpness, orientation and rotations.

B. Machine Learning Intoxicated Selfie Detection Pipeline

We synthesized a machine learning pipeline for intoxication detection, which had image pre-processing steps including face detection, locating facial landmarks and aligning faces, as well as feature extraction, feature selection and classification of faces as sober or drunk (Figure 3). We will expound on each of the steps in subsequent sections.

Figure 3: Selfie drunkenness classification pipeline.

C. Image Pre-Processing 1) Face Detection

In order to detect all the faces in an image, we used the Histogram of Oriented Gradients (HOG) method [14] with its main steps shown in figure 4.

Each image's colors and gamma values are first normalized. Then the local directions of gradients (intensity changes from light to dark) in the images are identified. After weighted voting and contrast normalization, the HOG patterns inside the detection window are compared to HOG patterns in a large dataset of faces. We used the HOG implementation from Dlib, a python image processing and machine learning library [19].

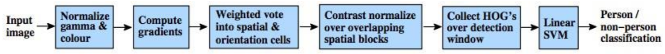

Figure 4: Detecting Faces in Images using the HOG [14].

2) Locating Facial Landmarks

Landmarks are parts of the face such as the mouth, eyes and nose that are likely to change meaningfully in response to alcohol or other stimulus. After detecting faces in the prior step, we then detected landmarks within the image using the Facial Landmarking algorithm [20]. Facial landmarking initially places 68 dots on the average location of facial features, which have been determined by a sampling of thousands of existing face images. Then, an iterative process morphs the shape of the points based on gradients in the image until the shape of the points matches the shape of the face. The algorithm maps landmarks to the face more accurately with every iteration (figure 5). We used the Dlib implementation of this facial landmarking algorithm [19].

Figure 5: Landmark estimates at different iterations of the facial landmark algorithm [20].

3) Aligning Faces using Landmarks

Using the facial landmarks found in photographs of faces, images in the dataset were rotated, centered and aligned. Alignment warps images so that landmarks occur at standard locations. In order to rotate and center the images the algorithm takes the center point between the eyes and then rotates, scales, and crops the image to make the images uniform (Figure 6). We used the Dlib implementation of the face alignment algorithm.

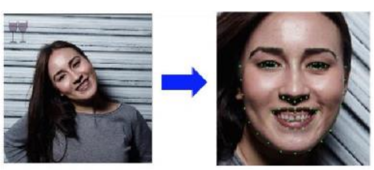

Figure 6: Example of the result of face alignment [18].

D. Feature Extraction and Exploration 1) Landmark positions

Our initial experiments used the X, Y locations of 68 facial landmarks of aligned faces as features, shown in Figure 7a.

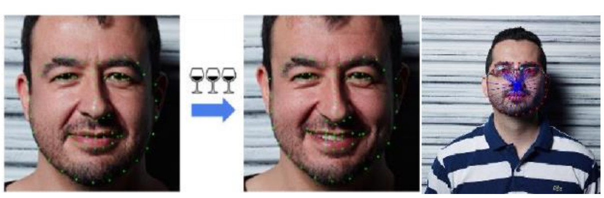

Figure 7: a) Landmark changes after three glasses of wine [18] and b) Landmark vector features

2) Landmark vectors

Inspired by prior work that utilized vectors of facial landmarks to detect emotion [21], we extracted vectors of facial landmarks as features. Landmark vectors were generated by drawing vectors from the center of the face (average value of the 68 landmarks) (Figure 7b). The distance between landmark points, and the angle of the vector in the aligned face, were also features.

3) Landmark lines

In order to extract more information about facial changes, we connected lines between landmark points and computed the distance between landmarks. These lines were to capture the effects of drunkenness on the subject, including changes to the shapes of the eyes, mouth, and cheeks. We also added angles of landmark lines in aligned faces as features.

4) Detecting facial lines and wrinkles using Canny Edge detection and the Hough transform

Intuitively, alcohol relaxes facial muscles and might change the wrinkles and lines of the face. We used the Canny edge detection algorithm to extract facial lines and wrinkles as features. First, a Gaussian filter is applied to the image, which effectively blurs the image to reduce noise [22]. Afterwards an intensity gradient of each pixel is determined for the image, which places the slope of each edge and the direction of every pixel within the image. Figure 8 below an input image, the image converted to grayscale, and a canny version of the input image. Once the edges were defined the image was further processed using a Hough transform, in order to determine the lines, present on the face. Figure 9 (right) displays an image in which the most prominent lines on a subject's face were highlighted using Hough transform.

Figure 8: Grayscale image of a subject's face (left) Image with canny edges (middle) image with Hough transform lines (right)

5) Forehead redness

The face gets more red when a subject consumes alcohol, especially their cheeks and forehead. We segmented out the subjects' forehead using a semantic segmentation algorithm, which we embellished to identify untestable foreheads that were typically covered by hats or hair. Our algorithm initially estimated the color and texture of the subject's face and inferred that an object was covering the forehead for regions that had a significantly different texture and color from the initial sample. Figure 9 shows the results of the forehead color detection step. We then extracted colors and features from a usable forehead segment to determine the redness of the subject's forehead.

Figure 9: Forehead Detection [18]

6) Smiles, Lips and Eyes

Subjects in the *3 Glasses Later* dataset tended to smile more after drinking, which caused some mis-classification. Smiling people who were not drunk were sometimes mis-classified as drunk. Sober images were generally more serious and less energetic. Segmenting out subjects' lips and eyes reduced the confounding effect of smiles. While accuracy was relatively unchanged, the number of false positives was reduced.

7) Feature engineering: normalization

Since each user might have different base levels of facial redness to begin with, we explored whether changes in facial redness were more predictive than absolute values of facial redness. To compute changes in facial redness, we subtracted values of color features in a subject's sober forehead image from color features of the same subject after drinking.

E. Image Augmentation

The collection of 212 photographs from the *3 Glasses Later* project was a fairly small dataset on which to build a robust machine learning classifier. Also, as the pictures were taken in a studio with controlled lighting, the dataset did not reflect many real world conditions users could potentially take photographs in. A classifier trained on such a dataset would generalize badly and have lower accuracy in the real world. Therefore, to simulate additional realistic conditions, we examined Internet images of selfie images at bars and parties. Inspired by the image characteristics we observed, we augmented the original dataset by transforming each original image to versions that might depict real world scenarios. Specifically, we found that bar selfies often had blurry images, low lighting, washed out/saturated from lighting, added tint/color by bar lights, rotated images and skewed faces.

1) Dataset augmentation using the imaug library

We used the *imgaug* python library to apply these modifications to our images [23]. We applied the following augmentations: image rotation, brightening, blurring, changing perspective, changing contrast, and adding tint. We used experimentally determined parameter values that accurately represent real life scenarios (listed under each image below in figures 10-12) [18]. We applied these augmentations to produce 11 additional images photos per original photo.

Fig. 10: a) Image rotation (-8 to 8 degrees). b) Add brightness to the image (45 to 45 brightness).

Fig. 11: a) Gaussian Blur (sigma=0 to 1). b) Changing perspective (50% chance of flipping horizontally).

Fig. 12: a) Perspective Transform (scale = 0 to 0.075). b) Adding tint (0 to 30% to Red, Green, or Blue).

IV. Results

We classified how many glasses of wine each subject had consumed (0,1,2,3). We also created 2 bins (target classes): "sober" (0 or 1 glass of wine) and "not sober" (2 or 3 glasses of wine). We used an 80/20 train/test split, k-fold cross validation (10 folds), compared various classifier types (figure 13) and generated confusion matrices.

A. Impact of Image Augmentation

Our augmented dataset contained 2,332 images which improved our models' ability to classify subjects' drunkenness. The accuracy of our models for classifying images into "sober" or "not sober" categories increased when using augmented images for every model we tested (figure 14). The standard deviation of accuracies between folds also decreased for most models. The addition of augmented photos to our dataset did not change the breakdown of false or true positive rates. Predicting sober was more accurate than predicting not sober.

B. Effectiveness of Various types of Features

We compared various classifier types including the Linear Support Vector Classifier, Polynomial SVC, Random Forest, and Decision Tree Classifier. Of these classifier types, Random Forest consistently performed the best. For this comparison (Figure 13) we also used subject-level splitting wherein each subject's data could only appear in either the training or test set, but not both. After extensive experimentation, we found the following features to be most predictive and used them in our final model 1) Landmark points – x, y 2) Landmark vectors – length and direction and 3) Landmark lines – distance. We also included normalized variants of these features (i.e. drunk feature value – sober feature value).

V. DrunkSelfie App

As a proof of concept, we created a prototype android app that could take photos and classify them using our machine learning intoxication classifiers. In order speed up classification, we processed images taken by the app using a remote Linux server. The server sent results back to the phone after intoxication classification. The app has a simple user interface, with a button for capturing a photo, and another for switching from front to rear camera (figure 15).

Implementing the DrunkSelife App: Figure 16 is an overview of the photo taking and classification process. The Android camera2 API is used to capture the photos. After a

photo is captured, it's uploaded to our server for processing. The Android AsyncTask and Java HTTP library is used for communicating with the server asynchronously.

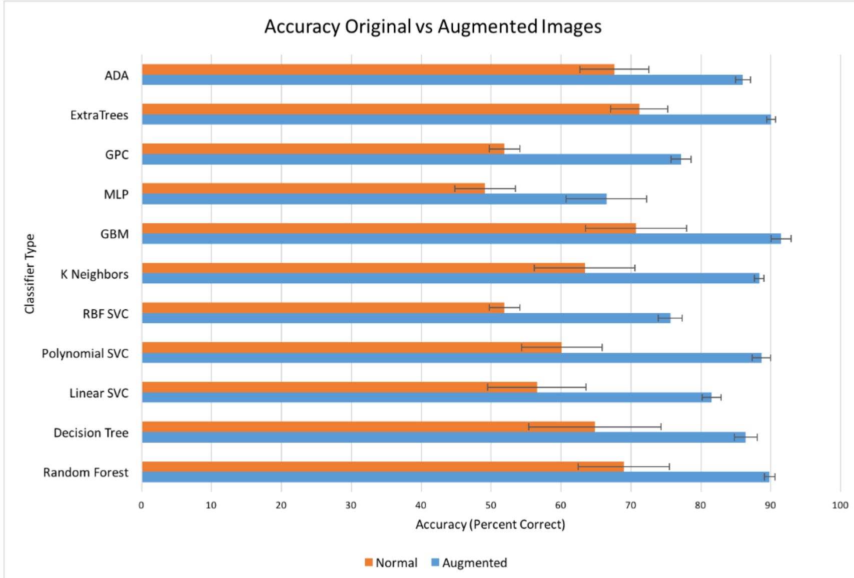

Figure 13: Accuracy of Augmented Images Sober or Not Sober

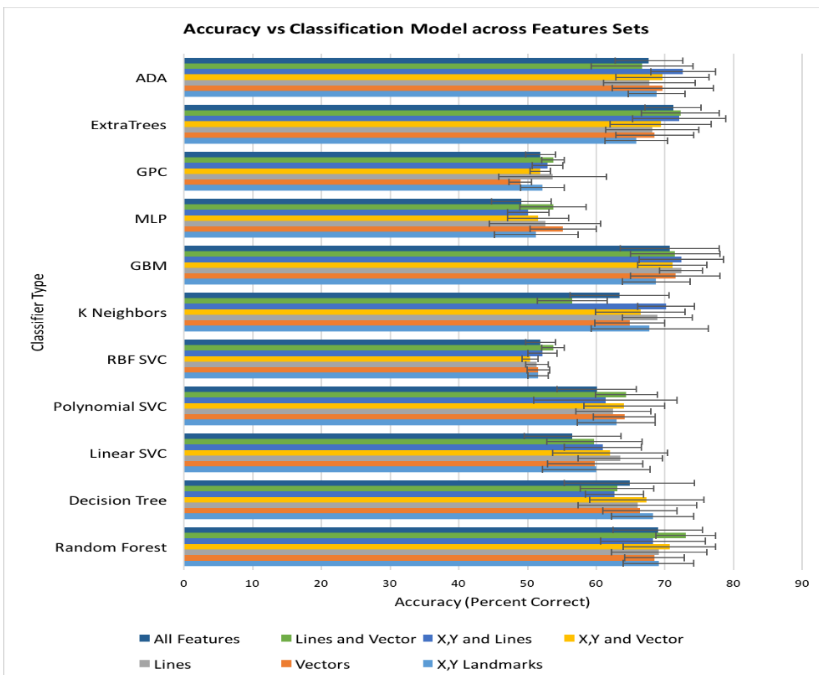

Figure 14: Classification Accuracies with various Features and Classifiers (with subject-level splitting).

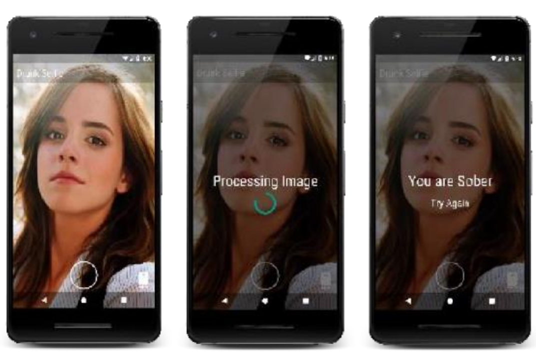

Figure 15: *DrunkSelfie* Prototype Android App

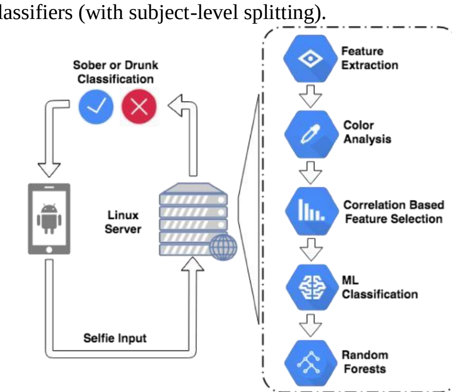

Figure 16: Architecture of *DrunkSelfie* Android App

VI. LIMITATIONS OF OUR WORK

While we achieved encouraging results, our work has several limitations that could be addressed in future work.

a) *Limited Dataset:* The *3 Glasses dataset* contained photos that were staged, captured in ideal studio lighting and not real world representations of drinking situations [18]. The images contained exactly one face. In real drinking situations, multiple drinkers might be in a single photograph. The subjects in the dataset were not diverse (mostly Brazilians). Some features such as facial redness might not work on subjects of a darker skin tone. Only wine was consumed by all subjects. Finally, using the number of glasses of wine consumed as a measure of intoxication is approximate as different people are affected differently by the same quantity of alcohol.

b) *Coarse Classification:* Currently our system does not accurately place each subject into the categories of drinking 0, 1, 2, or 3 glasses of wine, or estimate their actual BAC. In future, if we have access to a dataset in which subjects record their BAC, we can generate a model that classifies the BAC of drinkers.

c) *Smartphone Internet Access Requirement:* In Future, smartphones will have the processing power to perform image processing, feature extraction and classification on the device, eliminating the need for Internet access for our *DrunkSelfie* app.

d) *Comprehensive evaluation of DrunkSelfie app by users:* needs to be done including establishing user acceptance, attitudes, preferences and efficacy.

VII. CONCLUSION

Impaired driving accounts for 29 percent of all traffic-related deaths in the US [1]. A mobile app that can passively detect intoxication will assist users in making safe drinking decisions. We investigated whether smartphone user intoxication levels can be detected from facial photographs (selfies) taken while drinking. By analyzing 53 subjects in the *3 Glasses Later* intoxicated faces dataset, we were able to identify facial landmarks, vectors and facial structures as features that reliably indicate drunkenness. Gradient Boosted Machines classifiers had an accuracy of 81% at detecting when a subject was sober or drunk. Using our best classifiers, we created a *DrunkSelfie*, an Android app that detects intoxication from a selfie.

REFERENCES

[1] CDC Injury Center, "Impaired Driving: Get the Facts," *CDC Injury Center*, 2017 URL: https://www.cdc.gov/motorvehiclesafety/impaired_driving/impaired-drv_factsheet.html. [Accessd April 22, 2019].

[2] C.M. Steele and R.A. Josephs, "Alcohol myopia: Its prized and dangerous effects," *American Psychological Association*, vol. 45, pp. 921-933, 1990.

[3] T.K. MacDonald, M.P. Zanna and G.T. Fong, "Decision making in altered states: effects of alcohol on attitudes toward drinking and driving." *J.Pers.Soc.Psychol.*, vol. 68, pp. 973, 1995.

[4] J. Jaccard and R. Turrisi, "Cognitive Processes and Individual Differences in Judgments Relevant to Drunk Driving," *J.Pers.Soc.Psychol.*, vol. 53, 1987. pp. 135-145.

[5] Chuang-wen You, Kuo-Cheng Wang, Ming-Chyi Huang, Yen-Chang Chen, Cheng-Lin Lin, Po-Shiun Ho, Hao-Chuan Wang, Polly Huang, and Hao-Hua Chu. 2015. SoberDiary: A Phone-based Support System for Assisting Recovery from Alcohol Dependence. In *Proc* CHI '15.

[6] H. Wilson, S.R. Stoyanov, S. Gandabhai and A. Baldwin, "The Quality and Accuracy of Mobile Apps to Prevent Driving After Drinking Alcohol," *Jmir Mhealth and Uhealth*, vol. 4, pp. e98, July, 2016.

[7] Statista, "Share of adults in the United States who have ever taken a selfie as of February 2017, by age group," URL: https://www.statista.com/statistics/304861/us-adults-shared-selfie-generation/.

[8] BACtrack, "Three Types of BAC Testing," *BACtrack, 2017*. URL: https://www.bactrack.com/blogs/expert-center/35043461-three-types-ofbac-testing/. [Accessed April 22, 2019].

[9] DUI California, 2017 Available: http://www.dui-california.com/graphics/810fml11.jpg,. [Accessed April 22, 2019].

[10] C. Aiello and E. Agu, "Investigating Postural Sway Features, Normalization and Personalization in Detecting Blood Alcohol Levels of Smartphone Users" *Proc IEEE Wireless Health (WH) 2016*, pp. 1-8..

[11] C. You *et al*, , "CarSafe app," *in Proc MobiSys 2013*, pp. 13-26.

[12] G. Koukiou and V. Anastassopoulos, "Neural networks for identifying drunk persons using thermal infrared imagery," *Forensic Science International*, vol. 252, pp. 69-76, Jul. 2015.

[13] D. Hammerstrom, "Neural networks at work," *IEEE Spectrum*, vol. 30, pp. 26-32, 1993.

[14] Shashua A., Gdalyahu Y., and Hayon G., "Pedestrian detection for driving assistance systems, Single-frame classification and system level performance", Proc. of *IEEE Intelligent Vehicles Symposium*, 2004

[15] M. Hanselmann, *et al*, "Towards Digital Staining using Imaging Mass Spectrometry and Random Forests-Technical Report," 2009.

[16] A. Lanitis, C. Draganova and C. Christodoulou, "Comparing different classifiers for automatic age estimation," *IEEE Trans on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, pp. 621-628, Feb. 2004.

[17]] E. Makinen and R. Raisamo, "Evaluation of gender classification methods with automatically detected and aligned faces," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 541-547, 2008.

[18] M. Alberti, "3 Glasses Later," 2015. URL: https://www. masmorrastudio .com/wine-project. [Accessed Feb. 20, 2018].

[19] E. King Davis, "Dlib-ml: A Machine Learning Toolkit," 2009.

[20] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *Proc IEEE CVPR 2014*, pp. 1867-1874.

[21] P. van Gent, "Emotion Recognition With Python, OpenCV and a Face Dataset," April 01, 2016. Available: http://www. paulvangent.com/2016/04/01/ emotion-recognition-with-python-opencv-and-a-face-dataset/.

[22] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI8, no. 6, 1986. pp. 679-698.

[23] A. Jung, "imgaug," vol. 0.2.5, 2017. URL: http://imgaug. readthedocs .io/en/latest/index.html/. [Accessed April 22, 2019].

[24] B. Suffoletto, P. Gharani, T. Chung, and H. Karimi, Using phone sensors and an artificial neural network to detect gait changes during drinking episodes in the natural env, *Gait and Posture,* Feb 2018, pgs. 116-121

[25] A Mariakakis *et al*, Drunk User Interfaces: Determining Blood Alcohol Level through Everyday Smartphone Tasks", in Proc CHI 2018.

[26] D Santani *et al, D*rinkSense: Characterizing Youth Drinking Behavior using Smartphones, *IEEE Trans. Mobile Comp.*, Vol 17, No. 10, Oct 2018

[27] University of Notre Dame, "Blood Alcohol Concentration,", 2017. URL: https://mcwell.nd.edu/your-well-being/ physical-well-being/ alcohol/blood-alcohol-concentration/. [Accessed April 22, 2019].

[28] New iPhone App , intoxicheck™, Aims to Reduce Drinking and Driving, Takes Guesswork out of Judging Impairment URL: https://www. prnewswire. com/news-releases/new-iphone-app--intoxicheck-aims-toreduce-drinking-and-driving-takes-guesswork-out-of-judgingimpairment-136248573.html [Accessed April 22, 2019]