# Drunk Selfie Detection

Detecting drunkenness in photographs of faces

By:

Ian Banatoski

Paul Roberts

Colin Willoughby

Submitted To:

Professor Emmanuel Agu, Advisor

Date: April 24th, 2017

Major Qualifying Project Report submitted to the faculty of

Worcester Polytechnic Institute in partial fulfillment of the requirements of the

Degree of Bachelor of Science

*This report represents the work of WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its website without editorial or peer review. For more information about the projects program at WPI, please see http://www.wpi.edu/academics/ugradstudies/project-learning.html*

# Table of Contents

# Abstract

The goal of this project was to extract key features from photographs of faces and use machine learning to classify subjects as either sober or drunk. To do this we analyzed photographs of 53 subjects after drinking wine and extracted key features which we used to classify drunkenness. We used random forest machine learning to achieve 81% accuracy. We built an android application that using our classifiers to estimate the subject's drunkenness from a selfie.

# Acknowledgments

We would like to thank Professor Emmanuel Agu for guiding this project to its completion. We would also like to thank Marco Alberti for allowing us to use his *3 Glasses Later* photograph collection. Alberti's photography projects can be found on his website at www.masmorrastudio.com.

# 1 Introduction

In 2015, over 10,000 people died in motor-vehicle accidents related to alcohol-impaired driving. This accounted for "nearly one-third (29%) of the of all traffic-related deaths in the United States" [1]. Today, alcohol plays an important role in many social aspects of American culture. With twenty-four million Americans consuming an average of 74 alcoholic drinks per week [2], the effects of alcohol on society are hard to ignore. Many Americans gather in a social setting when drinking, typically at bars or in public gatherings. This aspect of drinking culture results in a need for transportation by the individuals involved. According to the 2014 American Community Survey Public Use Microdata Sample (ACS PUMS) from the US Census Bureau, 91% of Americans own at least 1 motor vehicle [3]. Therefore, for many people, the primary means of transportation to social drinking events is through the operation of a motor vehicle. This means that many people have to make the dangerous decision on whether or not to drive home after consuming alcohol. In addition to vehicles being the main form of transportation for many, smartphones are growing in popularity and becoming an increasingly common item to own in the United States. Smartphone's rising popularity made it an ideal platform to develop a tool to help aid users in making responsible decisions.

## 1.1 Alcohol's Effects on Perceived Intoxication

The effects of alcohol on the body (as it pertains to drinking and driving) are two fold in that you have reduced coordination and motor functioning, as well as impaired decision making [4]. Research done by C. M. Steele demonstrates that intoxicated people have a smaller scope of items they can attend to at once, and it's hypothesized that after drinking you're likely to only take into account the most prominent cues [4]. The decision to drive or not after drinking has many consequences that may be overlooked by someone who is drunk. These negative consequences of driving include the risk it poses to the safety of the driver and others around them, as well as the possibility of being charged with Driving Under the Influence (DUI). Unfortunately, the cons tend to get overlooked in the decision making process for more immediate positive cues, such as getting home quickly, or avoiding the hassle of paying for a ride home and having to pick up your car the next day. A sober person would likely weigh all the effects of their decision to drive, but alcohol often leads drinkers to make decisions quickly and often prevents people from understanding the possible consequences of their actions [5].

Figure 1 shows how people incorrectly perceive their level of drunkenness after several drinks over the course of 1, 2, and 3 hours. People begin to underestimate the amount of alcohol in their system even after as few as 2 drinks and many people forget that alcohol continues to be metabolized even after they stop drinking.

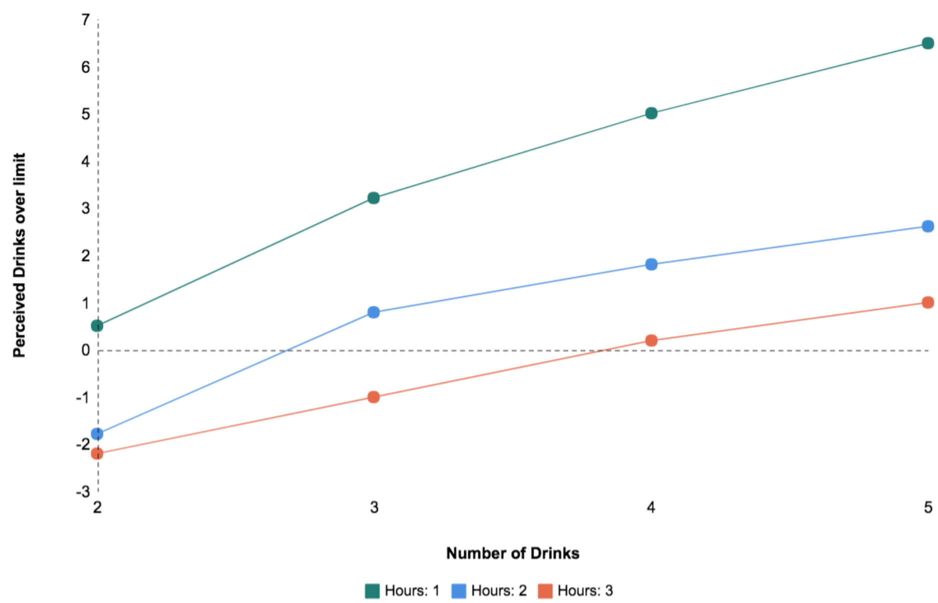



**Figure 1:** Mean Judgment of Drinks Over the Legal Limit [6]

Steele suggests an effect he calls *drunk invincibility* could contribute to the rate of drunk driving. Drunk invincibility refers to the tendency of someone to overestimate their ability to drive while intoxicated, and their general feeling of immunity to any consequences (physical or legal) that may result from the decision to drive [5]. Additionally, a study conducted by James Jaccard and Robert Turrisi looked into people's ability to estimate their BAC and number of drinks they've had, showing that between 30% and 40% of people would incorrectly guess they were below the legal limit to drive, when they were actually above the limit [6].

A common solution to drunk driving is assigning a member of a group to be a designated driver. Designated drivers are supposed to stay sober so the group can get home safely. Unfortunately, this is not always the case. The 2007 National Roadside Survey found that 13% of designated drivers had consumed some alcohol that day [7]. This leaves drunk passengers at risk from their designated drivers because there is no easy way for drunk passengers to determine the sobriety of their drivers.

## 1.2 Smartphone Usage

Smartphones can be used to provide millions of people with new tools to understand how drunk they are without requiring them to own separate hardware. As of November 2016, a Pew Research survey found that 77 percent of all Americans and 92 percent of those aged 18 to 29 years old owned a smartphone [8]. Smartphones are powerful tools that are highly accessible to the general public and more importantly, potential drunk drivers. Despite thousands of alcohol related smartphone applications being available today on the market, many of which try to tell users if they are sober enough to drive, most of these phone applications are inconvenient and difficult to use [9]. Therefore our team focuses on selfies, an already popular use of a smartphone for many people. According to a survey conducted in February of 2017, "87 percent of U.S. adults aged 18 to 34 years had taken a photograph of themselves and uploaded it to a social media website” [10]. Thus we believe an engaging and easy to use selfie application that can properly identify users’ drunkenness could become a part of everyday smartphone use and help prevent accidents caused by drunk driving.

## 1.3 Project Goals

People often have incorrect perceptions of their intoxication level and therefore drive drunk, potentially harming themselves and others. In order to protect the public from future risks posed by drunk driving, our project aims to reliably identify drunkenness using 2D photographs of faces. A person's facial characteristics change after consuming alcohol; the face reddens and facial structures relax. Understanding the meaning of these changes and how to identify them could be used to estimate a general level of intoxication. This process could also be integrated into an image taking smartphone application to provide a tool for drivers and passengers to better understand drivers' level of intoxication. To complete this goal we will create an estimation model to accurately classify the intoxication level of a subject. This will be accomplished by:

- Developing an image processing pipeline that will extract key features from the 2D photographs of a subject
- Determining which facial features most effectively classify a person's level of drunkenness
- Creating a drunkenness estimator to properly classify drunkenness using the extracted features

# 2 Related Works

## 2.1 Related Technology

### 2.1.1 BAC

Blood Alcohol Content (BAC) is a standard referring to the percentage of alcohol in a person's blood. It is calculated by measuring the weight of ethanol in 100 milliliters of blood, or 210 liters of breath [11]. BAC is the best indicator for the level of intoxication of a subject and how impaired they might actually be. Breath and blood tests are the two most common ways of measuring BAC, with urine tests being a third less accurate alternative when the other two methods are not available. Field tests typically use breathalyzers, a device that measures BAC in a person's breath, because they are portable and less intrusive than the other two methods. Breathalyzers are generally not as accurate as blood tests, but are still usually admissible in court as evidence in a DUI case. Currently all 50 states in the US have a legal limit of 0.08% BAC for operating a motor vehicle [12]. All of the current tools to determine sobriety are reliant on BAC.

### 2.1.2 BAC Calculators

Many organizations have developed tools to alert drivers that their BAC is over the legal limit to drive. These tools range from handheld breathalyzers to BAC calculator smartphone applications. A study from August 2016 identified over 2,900 alcohol related smartphone applications [9]. These applications are easily accessible to drinkers, but require information drinkers do not have access to and lack trials to determine effectiveness.

BAC Calculators are popular tools that can be used to determine how long a drinker must wait until they are safe to drive. A study by The Centre for Accident Research & Road Safety in Australia tested 44 different drunk driving prevention smartphone apps for accuracy and engagement [9]. They found that most of the apps used the Widmark BAC formula, Figure 2, that uses gender, weight, time passed, and ounces of alcohol to determine a subject's BAC level.

**Equation 11 8/10 version of Widmark's formula.**

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

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

**Figure 2:** Example Widmark Formula [13]

These applications typically overestimated the time needed to wait to drive safely. The study found that the prevention apps were, "not engaging" and that "none have as yet been tested in trials to determine their effectiveness in reducing drunk driving behavior" [9]. The Centre for Accident Research & Road Safety also found that many of the calculators depended on highly precise measurements of the number and type of drinks consumed [9]. Intoxicated individuals leaving bars and restaurants might not know the strength of the beverages they were consuming. The lack of engagement from these apps and the work required to properly determine one's BAC make these types of drunk driving prevention apps less effective.

### 2.1.3 AlcoGait

Another tool to detect drunkenness is AlcoGait, a mobile phone application that predicts a user's BAC using accelerometer and gyroscopic sensor data collected by a smartphone. AlcoGait focuses on nine different walking features to help estimate a user's BAC: skew, kurt, gait velocity, residual step time, band power, XZ sway, XY sway, YZ sway, sway volume. AlcoGait uses the J48 classifier to determine the accuracy of readings and these features to achieve an accuracy of 89.45% when detecting a user's BAC level. The J48 classifier is a simple C4.5 decision tree that creates a binary search tree that is "constructed to model the classification process" [14]. This method of feature processing yielded an accurate result using AlcoGait's normalized data. We used a similar feature selection method and classifier. We used features extracted from the photograph of a subject's face to create a classification model to identify the subject as either sober or drunk.

### 2.1.4 CarSafe

CarSafe is a car safety smartphone application developed by students at Dartmouth College to reduce the risk of accidents on the road. Instead of measuring a driver's intoxication level before driving, the app uses a combination of the front and rear cameras on a phone to monitor a driver's habits on the road, and alert them if they show patterns of bad driving. These habits include not looking at the road, lane weaving, and careless lane changes. This approach means drivers who may not have even used a drunkenness detecting tool will still be alerted of the risk they are causing. The project had moderate success, citing around 80% accuracy at measuring specific bad driving tendencies, and 60% at detecting drowsy driving [15].

## 2.2 Infrared/Thermal Imaging

Recent studies have shown thermal and infrared visioning can be used to detect alcohol intoxication with great success. Thermal cameras can display multiple signs of alcohol consumption in human faces. Blood vessels in the nose and eyes dilate, which increases heat in the face [16]. The heat signature of the eye itself changes with the consumption of alcohol and the sclera temperature rises [16]. These extracted features are processed using t-tests to determine the significance of the differences between the two different data sets. It does this by comparing the difference between the means of each data set, the larger the difference between the means, the larger the t-score is. A larger t-score suggests that the results are more repeatable. Research conducted using a t-test resulted in a confidence of over 99%, indicating thermal imaging is a very accurate tool in determining a subject's sobriety [16]. However, many smartphones today do not have thermal cameras and therefore thermal image processing has minimal reach to the public. While smartphones cannot read the changing heat signature of a subject's face, visible surface changes as a result of drinking could be used to help identify their level of drunkenness.

## 2.3 Facial Recognition

Neural networks are at the heart of most modern image recognition systems. Their natural fit for image recognition stems from their ability to take complex inputs and use patterns in order to give meaning to those inputs (often finding subtle relationships that humans might otherwise have missed) [17]. This is an important ability because it means programmers looking to find patterns in a dataset do not have to know the rules of their input / output relationship. Instead, the work of finding what parameters cause a specific output is offloaded to the neural network. Furthermore, after a neural network is trained, it's able to generalize its knowledge to other inputs that are roughly similar to the original input data. For our purposes, the ability to learn and generalize means we can take an existing dataset of drunk selfies, run the images through a neural network algorithm, and train the network to identify how many drinks a person has consumed (even if it has never before seen a picture of that person when they were drunk) [17].

In order to properly identify whether or not a person is drunk one of the first things we do is properly isolate regions of the face that contain the data. Image segmentation is the process of defining key areas of an image that will be focused on. It typically involves convolving an image by compressing its data and then de-convolving it back into the full size image. The Pulse-Coupled Neural Network (PCNN) model is a popular biologically inspired neural net used to segment images. It is a single layered, two dimensional, laterally connected network of pulse-coupled neurons [18]. Using PCNN to divide the face into regions will allow more precise feature extraction and help prevent noise from dirtying available data.

Once the face is segmented into regions, a number of other processing techniques can be applied to the image to help extract the features available. Common techniques include applying Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Support Vector Machines (SVMs). PCA is a technique that searches for directions in the data that have the largest variance and allows the subsequent conversion of that data to a set of linear uncorrelated variables. This facilitates the creation of predictive models generated from analyzing related datasets. In addition, LDA is used to take into account the labels of the training set data which increases the class separability and uses it to find linear combinations of features. SVMs classify data into two categories using regression analysis to separate data points. Models that use neural networks to classify images often require a large training dataset. We found that SVMs and traditional machine learning classifiers were the best tool to accomplish the task of using features to identify a subject as either sober and drunk [18].

## 2.4 Support Vector Machines

One of the tools used to build a model that classifies features is the use of Support Vector Machines (SVM). SVM is a supervised machine learning method used to place data into two categories [19]. As the model trains on data, it makes the distinction between the two categories as large as possible, see Figure 3.

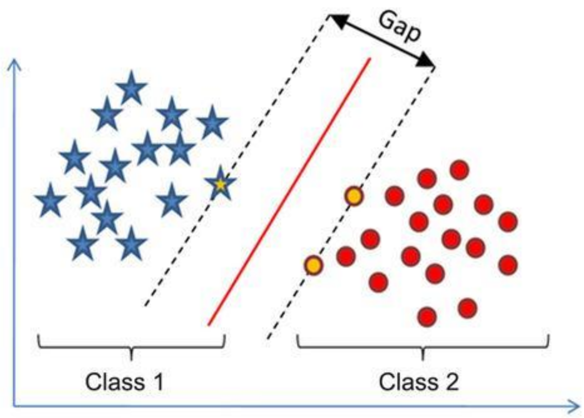



**Figure 3:** Example of SVM Clustering [29]

While the original version of the SVM algorithm only used binary classification, this model can also be applied to more complex, multi-class classification problems. We used SVMs to classify features based on number of drinks. Next, we observed how well the comparison of a single photo to those types of features indicated the number of drinks. The SVM machine learning algorithms used and additional algorithms we tested are described in further detail below.

### 2.4.1 Linear Support Vector Classifier

A linear support vector classifier is type of SVM that uses mathematical models to classify data points. Each data point is plotted in an N-dimensional space where n is the number of features of that space. These data points in our case belong to several different classes: sober, 1 drink, 2 drinks, and 3 drinks. Linear support vector classifiers use trends in the data to determine how to best segment each point into a solution space. The solution space is split using a hyperplane and the maximum margin best defines the output categories, see Figure 4.

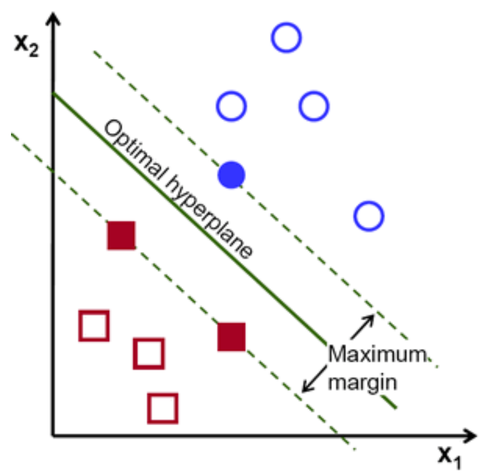



**Figure 4:** The graphical representation of a linear support vector classifier [20].

### 2.4.2 Polynomial Support Vector Classifier

A polynomial support vector classifier is a type of SVM that calculates a separation plane in higher dimensional spaces. Each data point is still plotted in an N-dimensional space. However, classification can be more accurate because it fits more directly to the data. This is more versatile than linear support vector classifiers because it allows the classification of data that is not easily separable using two dimensional hyperplanes.

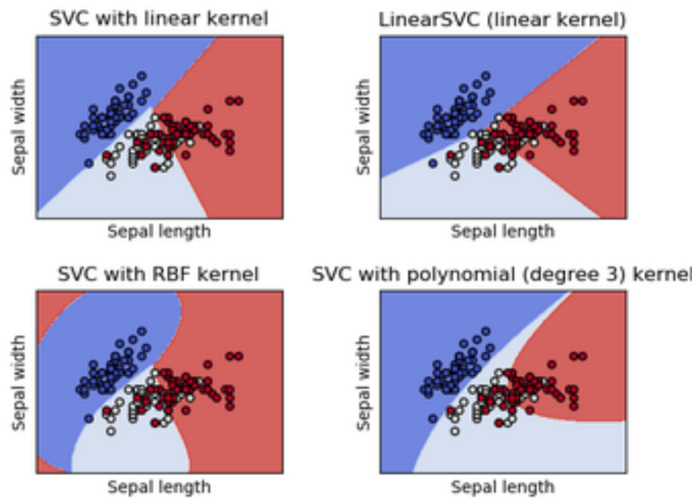



**Figure 5:** Types of Support Vector Classifiers [21].

### 2.4.3 Multilayer Perceptrons

We used the Multilayer Perceptrons (MLP) model to see if adding layers to our models improved our accuracy with features. MLP is a supervised learning algorithm that uses neural networks with at least three layers to classify inputs [22]. MLP learns using back-propagation of training data, comparing errors between neurons in the the neural network [19]. One advantage of MLP models is that they predict data that is not linearly separated, meaning that differences between classes could have complex clustering patterns with only one distinguishing metric, which can be clustered using multiple similar metrics [19]. We used SciKit's implementation of this learning algorithm written by Malcolm Ware to determine drunkenness from our facial features [22].

### 2.4.4 Random Forests

We used Random Forests learning to determine the effectiveness of decision based tree models on our features. Random Forests is an implementation of the Decision Tree learning method that uses randomness to prevent overfitting of the model [23]. Decision Tree algorithm continually divides the input space into subspaces. It does so by identifying lines, similar to the linear support vector classifier, and when it identifies the distinct regions within the data, it classifies those regions as separate categories.

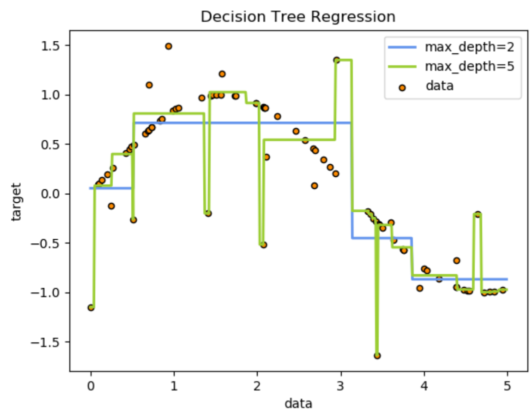



**Figure 6:** Decision tree with variable depths visualized [24].

Random Forests differs from the Decision Tree classifier in that it creates a set of decision trees based on the random selection of data-points, or subspace, from the training set. It selects these random samples and then generates a decision tree using the defined feature set. The algorithm makes a tree whose branches lead to classifications: no drinks, 1 drink, 2 drinks, 3 drinks. For each feature value, a decision is made to follow a certain path down the tree. A given combination of feature values will lead to certain output classification. Having several different decision trees allows for different variations of the main classification and builds a model less prone to noise. This is a robust way to classify data that isn't necessarily classified accurately using linear support vector machines, however it tends to overfit the test features. Random forests takes averages of decision trees trained on parts of the tests data. Our implementation of Random Forests used Sci-Kit's implementation of UCLA Professor Leo Breiman's Random Forest algorithm [23].

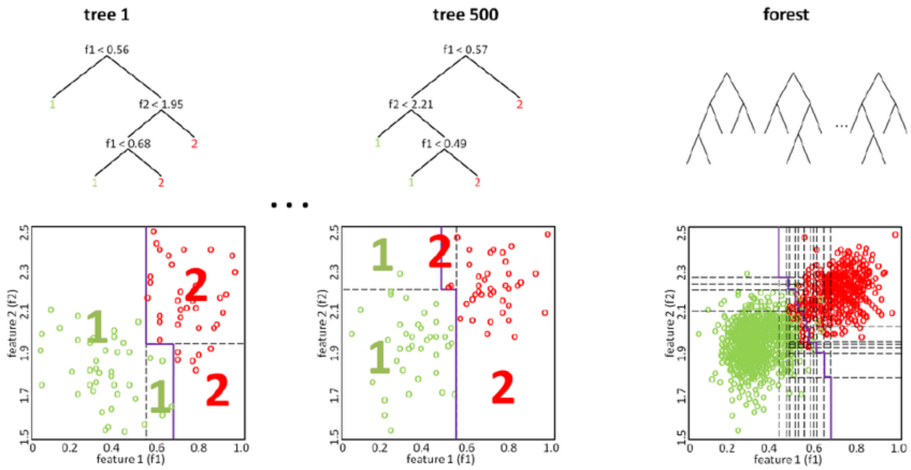



**Figure 7**: Random Forest diagram [24].

## 2.5 Smartphone Cameras

Most smartphones come with front facing cameras that allow users to take photos of themselves (selfies). An analysis of Instagram's public content and the tags "#me" and "#selfie" found that as of August 2017 there were over 600 million selfies posted. Google analyzed and tagged user's Google Photos libraries using machine learning and identified over 24 billion selfies [25]. Selfies are a common part of everyday life for smartphone users. With ever improving camera technology in the latest smartphones, most people can easily take high quality selfies from which drunkenness could possibly be detected.

In September 2017, Apple announced their flagship smartphone the iPhone X would use a facial unlocking feature that took advantage of an array of cameras and sensors at the front of the phone. One of the new camera's is an infrared camera. Past research on detecting drunkenness with infrared has been highly successful. Depending on how available Apple makes their new sensors to developers, this new sensor could make detecting drunkenness more reliable.

# 2.6 Case Studies involving Detection from Images of Faces

## 2.6.1 Case Study: Age Detection

Researchers have tested various facial feature comparison models that estimate the age subjects in photographs. A 2004 study by researchers Andreas Lanitis, Chrisina Draganova, and Chris Christodoulou titled, “Comparing Different Classifiers for Automatic Age Estimation” extracted features from photos and coded them into “face parameters” that could be used in an age estimator model to get an estimated age, see Figure 8 [26].

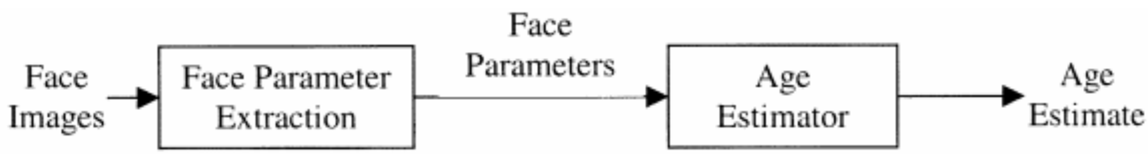



**Figure 8:** Block diagram of age estimation approach [26]

The study used four methods for estimating ages: Quadratic Functions, Shortest Distance Classifiers, Supervised Neural Networks, and Unsupervised Neural Networks. Quadratic Functions were used to make an optimized equation that finds an age based on the face parameter inputs. This approach identified ages within 5.04 years of error (Table 1). The Shortest Distance Classifiers estimator uses the distribution of the parameters for certain ages to classify which age a new set of facial parameter corresponds with. The accuracy of the estimator was worse than the Quadratic Functions approach with an error of 5.65 years. The Supervised Neural Networks approach used Multilayer Perceptrons (MLPs) and a learning algorithm to estimate the ages from training sets of face parameters. MLP estimated ages with an error of 4.78 years. The Unsupervised Neural Networks estimator used the Kohonen Self Organizing Map (SOM) neural network to estimate ages. SOMs build a map of the inputs during training which is then used to map new inputs to an estimated age. This approach found ages with an error of 4.90 years. The study added more steps to improve these estimated age predictions with age and appearance specific classifiers. This allowed them to improve the Quadratic Functions to predict age with an error of 3.82 years and MLP within 4.38 years.

**Table 1:** Results of Age Estimation Experiments (Error in years) [26]

| Method | Quadratic | Shortest Distance | MLP | SOM |
|---|---|---|---|---|
| Single Step | 5.05 | 5.65 | 4.78 | 4.9 |
| Appearance and age classifiers | 3.82 | 4.92 | 4.38 | N/A |

Age detection studies are using changes in facial features to classify and identify ages based on faces. Our project is attempting to achieve a similar goal by finding sobriety based on changes in facial features. The age estimators used known ages of the subjects in photographs to build out a prediction model. We hypothesize that if we have known levels of drunkenness, similar machine learning classification models can be built. Estimating age from faces varies from estimating drunkenness because the facial structure and size of faces change as subjects get older. Nevertheless, we can still use features similar to those used in this age detection study to classify drunkenness, such as the vectors created by landmarks recorded on a face.

### 2.6.2 Case Study: Gender Detection

A study conducted by Erno Makinen and Roope Raisamo aimed to determine which classification and face alignment methods would produce the most accurate detection of gender in images [27]. To begin, they implemented two different methods of face alignment: *Active Appearance Model (AAM)* based face alignment, and the *Profile Alignment* method. AAM uses a series of landmarks on and around the face in order to align and resize the image, while Profile Alignment creates intensity profiles to determine eye location, and aligns based on their relative position. Next, the aligned faces were passed through four different types of gender classification methods: A neural network, Support Vector Machine with two types of input modes, and Discrete

Adaboost with Haar-like features. Further details on the implementation of each of these methods can be found in their paper "Evaluation of Gender Classification Methods with Automatically Detected and Aligned Faces". The full list of variables can be seen below in Figure 9 (left) below, along with their classification accuracies (right) [27].

**Table 2:** Gender Classification Variables [27]

| Variable | Conditions |
|---|---|
| Gender classification method | SVM with LBP features<br>Neural network with face pixels<br>SVM with face pixels<br>Adaboost with haar-like features |
| Alignment method | None<br>Manual<br>Profile<br>AAM with eyes<br>AAM with eyes and nasal spine<br>AAM with eyes and mouth |
| Input image size[1] | 24*24<br>36*36<br>48*48 |
| Timing of alignment[2] | Alignment before resizing face image<br>Alignment after resizing face image |

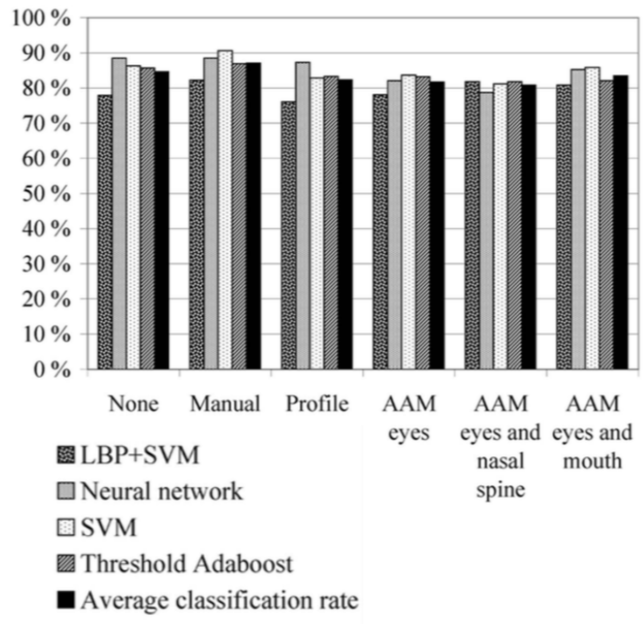



**Figure 9:** Gender Classification Rates [27]

The results in Figure 9 show that in practice, the SVM (pixel based input), Neural Networks, and Threshold Adaboost perform similar to each-other. It can also be seen that the addition of automatic image alignment did not increase the classification accuracy. Makinen and Raisamo concluded that, if you had to choose a single best-performing method, AAM alignment and SVM with pixel based input received the best overall classification scores, but they also noted that a better face alignment algorithm could help increase the rate of successful classification. The full list of alignment variables can be seen in Table 2. Given the success of SVMs at classifying gender, we decided to test several SVM models to determine their effectiveness at classifying drunkenness [27].

# 3 Methodology

## 3.1 Sources of Drunk Photographs

In order to test and develop our tool to detect intoxication we used a dataset of photos of people's faces with various levels of known drunkenness. Our preferred dataset had images of a person sober and then images of them with varying levels of intoxication. Using this data we trained our model to detect the difference in facial features as the subject consumes more alcohol.

### 3.1.1 Three Glass Later

Brazilian photographer and artist Marco Alberti captured photos of 53 different people while sober and then after drinking one, two, and three glasses of wine [28]. The project, titled *3 Glasses Later*, aimed to show the change in persona as people drank and had an enjoyable evening (see Figure 10). This dataset was not collected scientifically and may over dramatize the subject's emotions to make the art piece more moving. However, it does give clear photos of people as they drink with good lighting, framing, and quality. This data was used to help test and train the accuracy of our tool, but is not usable in a scientific way. While the uniform capture conditions of all the photos may be helpful in early stages of our tool, real world photos will have off-center faces, different angles, lighting, and other factors not found in this dataset [29]. Thus we utilized image augmentation to expand our dataset with images that reflect more real world conditions.

**Figure 10:** Example Photograph by Marco Alberti from *3 Glasses Later* Collection [28].

### 3.1.2 Social Media

Social media is another source of photographs of people who are drunk. There are thousands of photos of people tagged or labeled with the phrase “drunk selfie” on services such as Flickr, Imgur, Twitter, Instagram, Facebook, and Google. These photos vary in lighting, framing, and quality making them potentially better for training our sobriety estimator for the real world [29]. However, some of the photos aren’t of people’s faces and there is no way to confirm how drunk the subjects are in the photos. Additionally, these photos are also not paired with the sober photo of the subject making determining change between photos difficult. Perhaps the accounts of subjects could contain other non-drunk photos for comparison, but verifying they were sober would be difficult. Therefore we decided against testing using photographs from social media platforms.

## 3.2 Goals

The primary goal of our project was to develop a tool to reliably classify drunkenness in photographs of faces. To achieve our goal, we focused on the following objectives:

1. Develop an image pipeline that extracts key features that indicate drunkenness
2. Determine which facial features most effectively classify a person's drunkenness
3. Create a drunkenness classifier to accurately identify sobriety using the extracted features
4. Compare classifiers to optimize classification accuracy

To accomplish these objectives, we used a variety of methods, including implementing facial alignment, facial landmark detection, and pigmentation analysis outlined in detail below. Our approach yielded an understanding of which facial features are most useful in determining a person's drunkenness.

## 3.3 Our Machine Learning Intoxication Detection Pipeline

To accurately extract features from the data provided by Marco Alberti's *3 Glasses Later* dataset we developed a pipeline of image operations [28]. This pipeline of image operations detected and aligned faces, extracted features, and determined which features best indicated drunkenness. Finally, it used those features to classify the subject as either sober or drunk, see Figure 11.


**Figure 11:** Diagram of drunkenness classification pipeline.

### 3.3.1 Face Detection

In order to identify facial features that indicated drunkenness we first needed to find the faces in the photographs. A 2005 article by Navneet Dalal and Bill Triggs describes the Histograms of Oriented Gradients (HOG) method for detecting people in images [30].

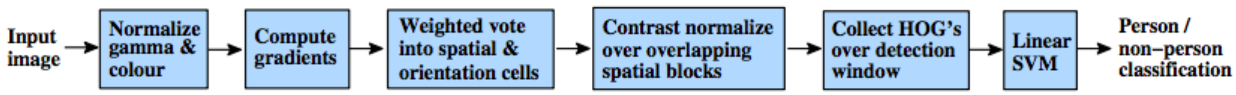



**Figure 12:** Identifying Subjects using HOG [30].

Figure 12 shows the steps required to detect a person in a photograph using HOG. The photograph is first converted to a normalized level of color and gamma values. Then the direction of gradients, areas going from light to dark, are identified. Once the gradient directions have been identified, they can be compared to other HOG patterns from a large dataset of faces. We use an implementation of the HOG algorithm from a python image processing and machine learning library called Dlib [31].

### 3.3.2 Locating Facial Landmarks

Once the subject's face had been identified in an image, we attempted to find and label areas of the face such as the mouth, eyes, nose, and jawline. We used a popular method of identifying certain areas of the face developed by researchers Vahid Kazemi and Josephine Sullivan in 2014 called Facial Landmarking [32]. This algorithm starts by placing 68 dots on the average location of facial features that 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 (error determined by distance of the shape from the gradients the algorithm is tracking). This process, as shown in Figure 13, maps the landmarks to the face more accurately with every iteration. We again used an implementation of this algorithm from the Dlib library [31].

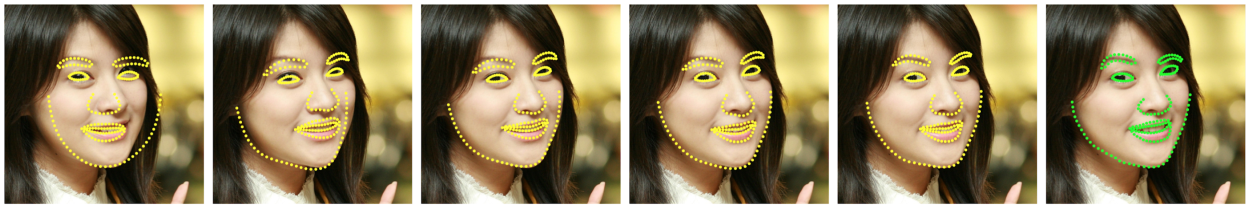


**Figure 13:** Landmark estimates at different iterations of the algorithm by Kazemi and Sullivan [32]

### 3.3.3 Aligning Faces using Landmarks

Using the facial landmarks found in photographs of faces we were able to rotate, center, and align the faces to a more standard form. To align faces that are slightly turned we warped the images based on their landmarks in order to fit standard landmark locations. In order to rotate and center the images the algorithm takes the centerpoint between the eyes and then rotates, scales, and crops the image to make the images uniform. An example of this operation can be seen in Figure 14. Aligning photos allows us to have a standard input of faces to be used in our drunkenness estimator model. We used an implementation of the face alignment algorithm from an OpenCV utility package called IMutils [20].

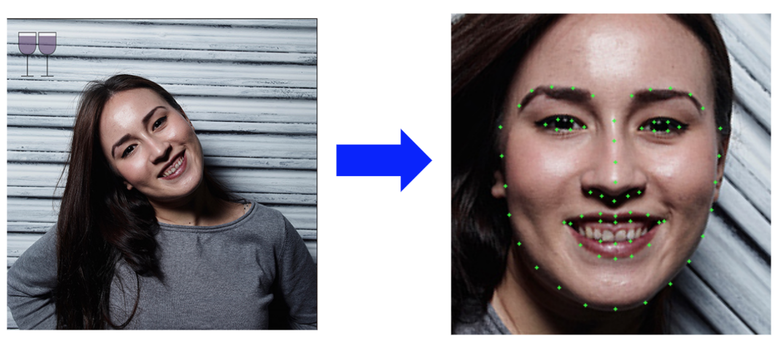


**Figure 14:** Face Alignment Result [28].

## 3.4 Features

### 3.4.1 Landmark Positions

Our initial experiments used a combination of X, Y positions of 68 facial landmarks of aligned faces as features, shown in Figure 15. These landmarks are considered to be points on the face that show the most significant changes. Facial landmarks are key points on the face that express position of the follow parts of the face:

- Mouth
- Right eyebrow
- Left eyebrow
- Right eye
- Left eye
- Nose
- Jaw

We found landmarks in photographs using the Dlib library's built in facial landmark detection tool. This tool's implementation comes from Vahid Kazemi and Josephine Sullivan paper *One Millisecond Face Alignment with an Ensemble of Regression Trees* (2014) [32]. We then looked at changes in facial structure and shape as potential features that indicate drunkenness. We visually analyzed photographs of sober and drunk subjects and noted that slight changes in facial landmark locations did exist. These changes were difficult to quantitatively measure by visually looking through the images, but were detectable by machine learning classifiers.

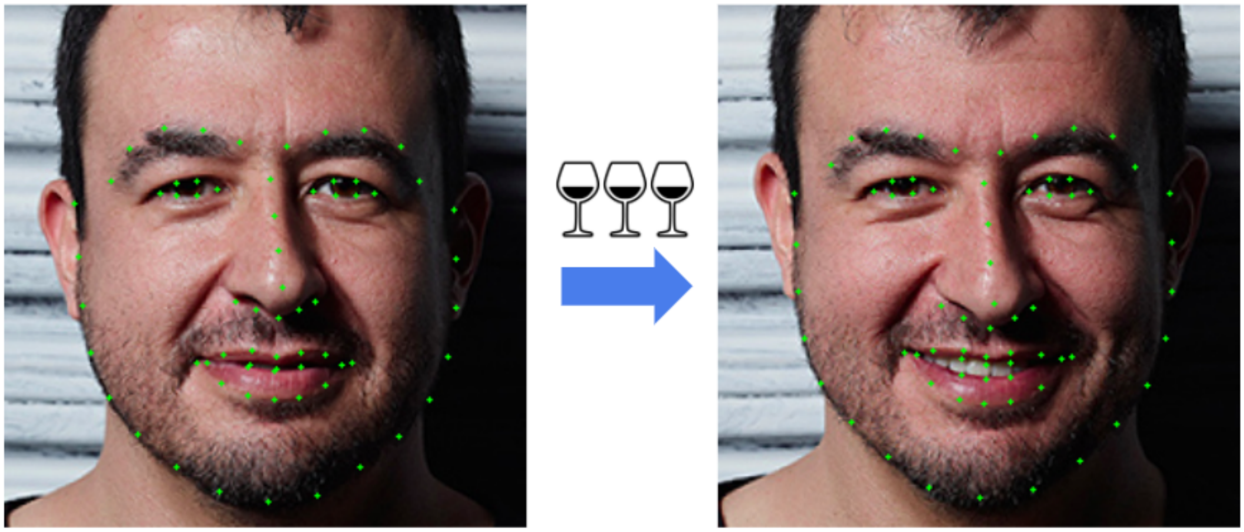


**Figure 15:** Landmark changes after three glasses of wine [28]

Using only the X and Y values of each point as features, we were able to classify images correctly as sober or not sober 69 percent of the time (as compared to 50 percent if guessing). A confusion matrix is the table that reveals how often a classifier predicts a label for a specific category. Each column represents the prediction categories that our model used and each row represents the real-world labels for the values being tested. Confusion matrices allow us to understand where our model is incorrectly identifying the alcohol consumption levels. The confusion matrix from one fold (Figure 16), shows the breakdown of how well the model classified the subject's drunkenness and which classes are being misclassified as other classes.

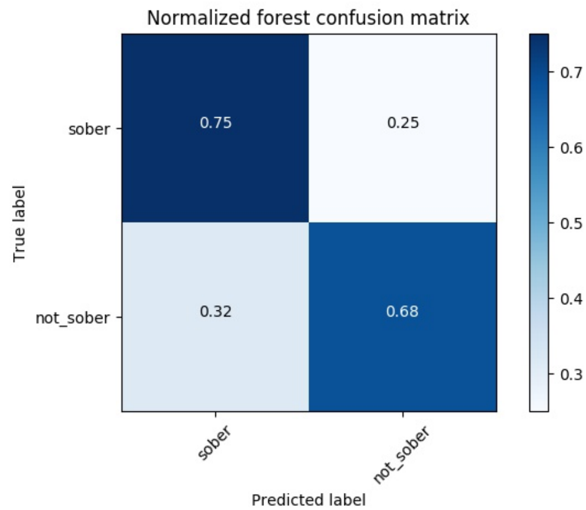



**Figure 16:** Confusion Matrix for the Random Forest Classifier using Landmark Positions as Features Confusion Matrix.

### 3.4.2 Landmark Vectors

In order to improve our classification models, we needed features that better indicated drunkenness. Since we were attempting to detect changes in faces as alcohol was consumed we focused on reviewing past studies that detected changes in facial structure. One prior experiment used a vectors generated from facial landmarks to improve accuracy in emotion detection [33]. We implemented this feature set by drawing vectors from the central point on the face (the average of the landmarks' x, y positions to each of the 68 landmarks) (Figure 17). The distance between the points and the angle of the vector, relative to the aligned face, are also both used as new features.

**Figure 17:** Vector Features [28]

We found that the vectors achieved about the same level of classification accuracy as landmarks at predicting drunkenness correctly and that using both increased the classification accuracy by around two percent for most classifiers (see Figure 18). Figure 35 shows a similar breakdown of accuracies in the confusion matrix for vector features as in landmark positions.

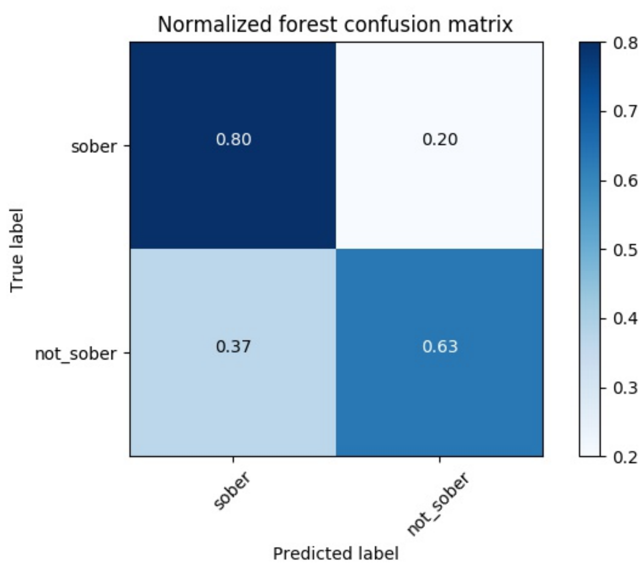



**Figure 18:** Confusion Matrix for the Random Forest Classifier using Vector Features Confusion Matrix

### 3.4.3 Hough Transform Line Detection

To increase the accuracy of our classifier we decided to test the detection of lines expressed by wrinkles in the face. Two of the most common ways to identify wrinkle lines on a subject's face is through canny edge detection and the other is through gabor filters. Canny edge detection is an edge detection algorithm developed by John F. Canny. For the algorithm to produce an image that displays only the most prominent lines, first a gaussian filter is applied to the image, this effectively blurs the image to help reduce any outlying noise [34]. 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 19 below shows the input, an image converted to grayscale, and the output, a canny version of the input image.

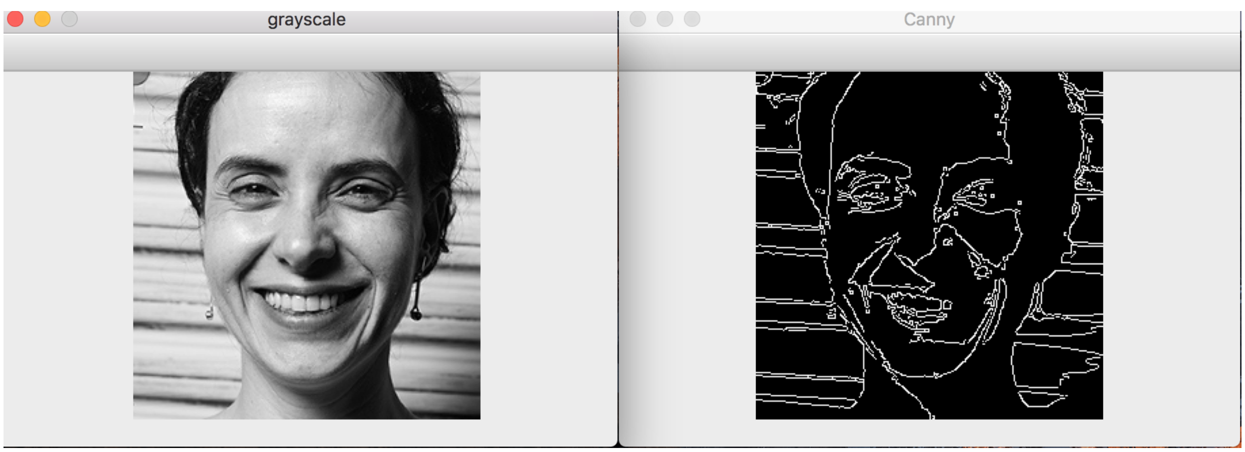



**Figure 19:** Grayscale image of a subject's face (left) and the corresponding canny version of the image (right).

Once the edges were defined the image was further processed using a hough transform, defines the shapes that are present within the image. In our case we apply a hough transform to the canny image to determine the lines present on the face. Figure 20 displays an image in which the most prominent lines on a subject's face were highlighted using hough transform and then overlaid on the original image.

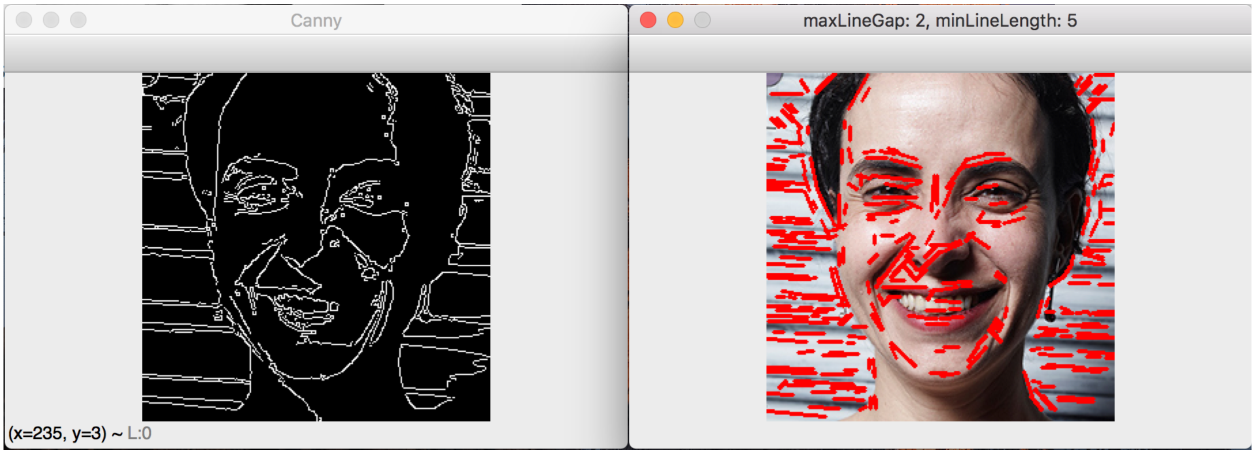



**Figure 20:** Input canny image (left). The detected lines using the hough transform are overlaid on the original image (right).

Upon further analysis we found that the lines created using a hough transform did not provide a consistent representation of the subject's facial structure. In addition we found that analyzing the data generated by recording the wrinkles in the subject's face did not match the necessary inputs required to test the accuracy of the classifier. Therefore, we decided against including this in our feature set and decided to use the lines formed by connected landmarks.

### 3.4.4 Landmark Lines

In order to extract more data about how changes occur using the landmark points, we connected lines between landmarks to outline certain regions of the face. This line feature set focused on the distance between landmarks and how that distance is affected by the drunkenness of the subject. We specifically built these features using inferences on sections of the face that we believed would change the most, for example the shapes of the eyes, mouth, and cheeks. We also added the angle of the lines relative to the aligned faces as a feature. Adding all of these features improved our overall accuracy by around one percent. As shown below in Figure 21, we found that using strictly landmark lines yielded an result of 75 percent accuracy.

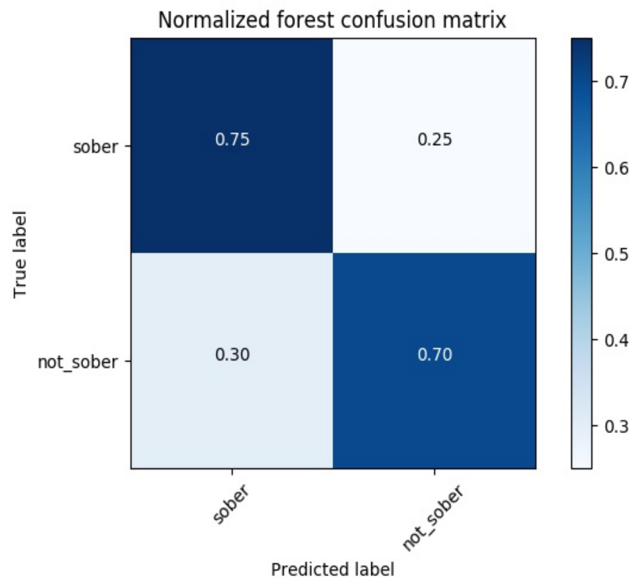



**Figure 21:** Confusion Matrix for the Random Forest Classifier using Landmark Line Features

### 3.4.5 Forehead Redness

To detect the redness of a person's forehead, we began by segmenting the forehead from the rest of the face. Our algorithm selected the section of the forehead in between a person's eyebrows, which we found to be the best location to find a representative sample of forehead color and also the location of the face least likely to be covered by hats or accessories in the *3 Glasses Later* photos [28]. Using the color and texture average on that sample, the algorithm then expanded the detected forehead area. Then the algorithm used the initial color and texture values as a way to threshold out sections that were likely to be an object covering the forehead (sections that have texture and color that are significantly different than the initial sample). In order to detect cases where the entire forehead may be covered by hair or an accessory, the algorithm rejected photos that had forehead colors that differ by a significant amount from the subject's nose. The results of this step in the forehead color detection can be seen in Figure 22: the first two photos are marked with a red "x" by the algorithm since it determined the foreheads shouldn't be used, and the second two photos show a blue highlight on the area of accepted pixels by the algorithm.

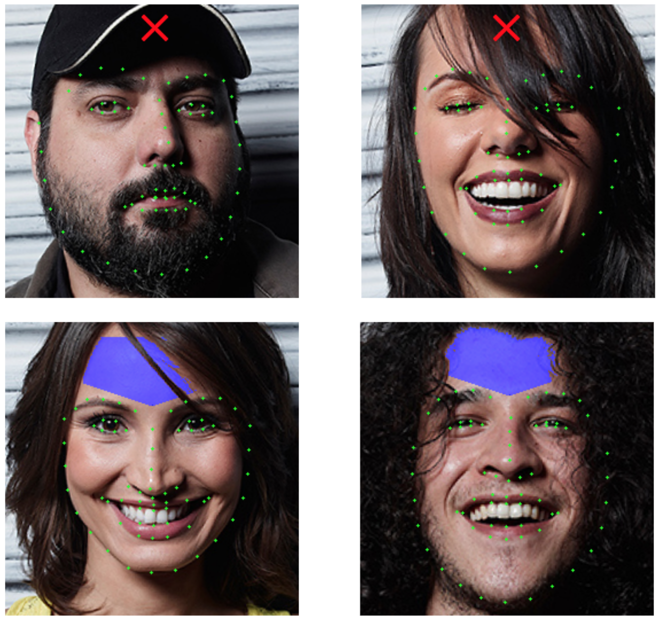


**Figure 22:** Forehead Detection [28]

After an acceptable forehead area has been established, we then extracted color data to determine the redness of someone's forehead.

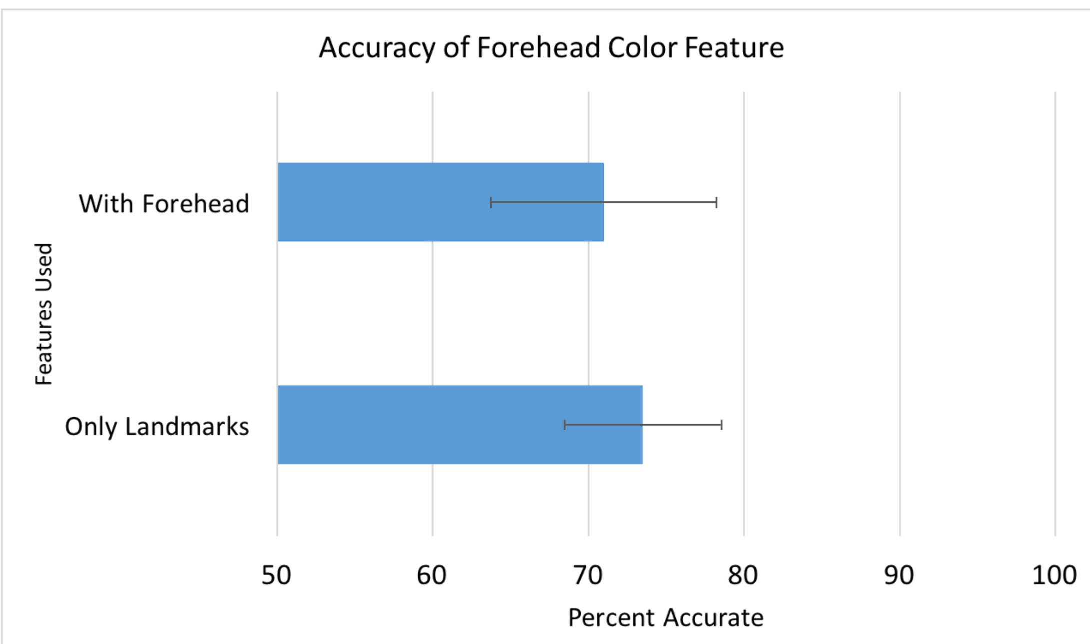



**Figure 23:** Accuracy of Classifier using Forehead Color

Unfortunately, after using forehead redness as a feature for indicating drunkenness, we found that redness in facial features did not significantly improve the accuracy of the classifier, see Figure 23. In addition, this made testing our models slow and also meant that we could not use this feature in certain photographs in which the subject's forehead was obstructed. These results led us to exclude color redness from our set of features and instead we focused on other features when testing our models.

### 3.4.6 Lips and Eyes

After testing one of the models on photos in the wild, we found that smiling tended to cause the model to incorrectly classify someone as drunk when they were sober. After analyzing the *3 Glasses Later* dataset, we found that many subjects smiled more after drinking [28]. Since our dataset is not exactly representative of the real world we decided to test accuracy when areas of the face such as eyes or lips were removed from the landmark and vector feature sets. As shown in Figure 24, removing these areas only reduced our accuracy slightly on our test results, but when experimenting with images outside our dataset, it appeared to slightly reduce false positives caused by smiling.

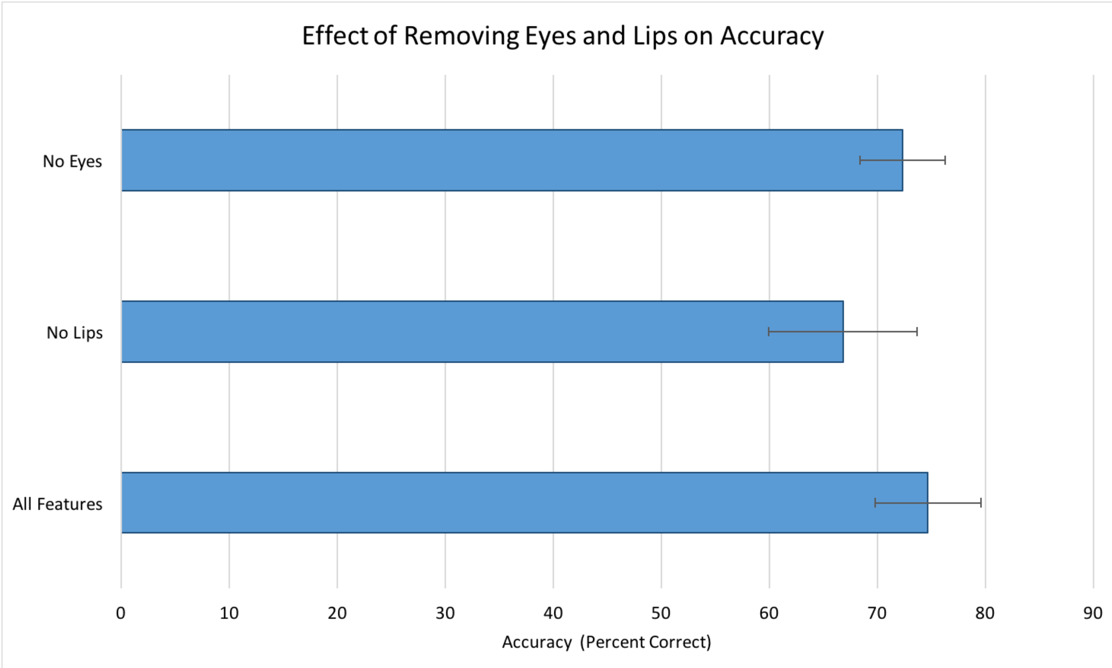



**Figure 24:** Accuracy of Random Forest after Removing Lip and Eye Features

This effect reflects one of the many issues with the *3 Glasses Later* dataset; the lack of emulation of real world subjects. The images posted by the artist were almost intentionally trying to show off changes as people drink more after a stressful day. Laughter and smiles were commonly seen in the 2 and 3 glasses photos, while the sober images were commonly more serious and less energetic. Since this dataset was our most comprehensive set of drunk photos, these biases played a role in affecting our model and making it less accurate in the real world [28].

## 3.5 Image Augmentation

The collection of 212 photographs from the *3 Glasses Later* project was a fairly small data set for our initial experiments. The dataset also did not reflect many real world photographs users could potentially take. Therefore to simulate additional realistic conditions in which people take selfies when going out to bars or parties, we examined examples on the internet of people taking selfies at bars. Finally, to increase our dataset size we altered our images to give them characteristics we observed in those selfies. Common characteristics identified from bar selfies:

- Blurry images
- Low lighting
- Washed out from lighting
- Added tint / color from bar lights
- Rotated image
- Skewed face

### 3.5.1 Implementing in Python with imgaug

We used an open source python library called *imgaug* in order to apply these modifications to our images. *Imgaug* is a python library with many common image augmentations implemented, allowing users to select what kind of augmentations they would like to make to photos in your dataset [35]. Some of the augmentations supported by *imgaug* include image rotation, brightening an image, blurring an image, changing perspective, transforming perspective, and adding tint. In our case each of these alterations were applied at a selected intensity in the specified range, which we determined would accurately represent real life scenarios (listed under each image below in figures 25-30) [28].

**Figure 25:** Image rotation (-8 to 8 degrees).

**Figure 26:** Add brightness to the image (-45 to 45 brightness).

**Figure 27:** Gaussian Blur (sigma=0 to 1).

**Figure 28:** Changing perspective (50% chance of flipping horizontally).

**Figure 29:** Perspective Transform (scale = 0 to 0.075).

**Figure 30:** Adding tint (0 to 30% addition of Red, Green, or Blue).

We applied these augmentations to our base images to produce 11 new additional augmented photos per original photo, each can be used as new data points. Our augmented dataset contained 2,332 images which improved our models' ability to classify subjects' drunkenness. The accuracy of our models to classifying images into Sober or Not Sober categories when training and testing on augmented images increased for every model we tested and the standard deviation of accuracies between folds decreased for most models, see Figure 31.

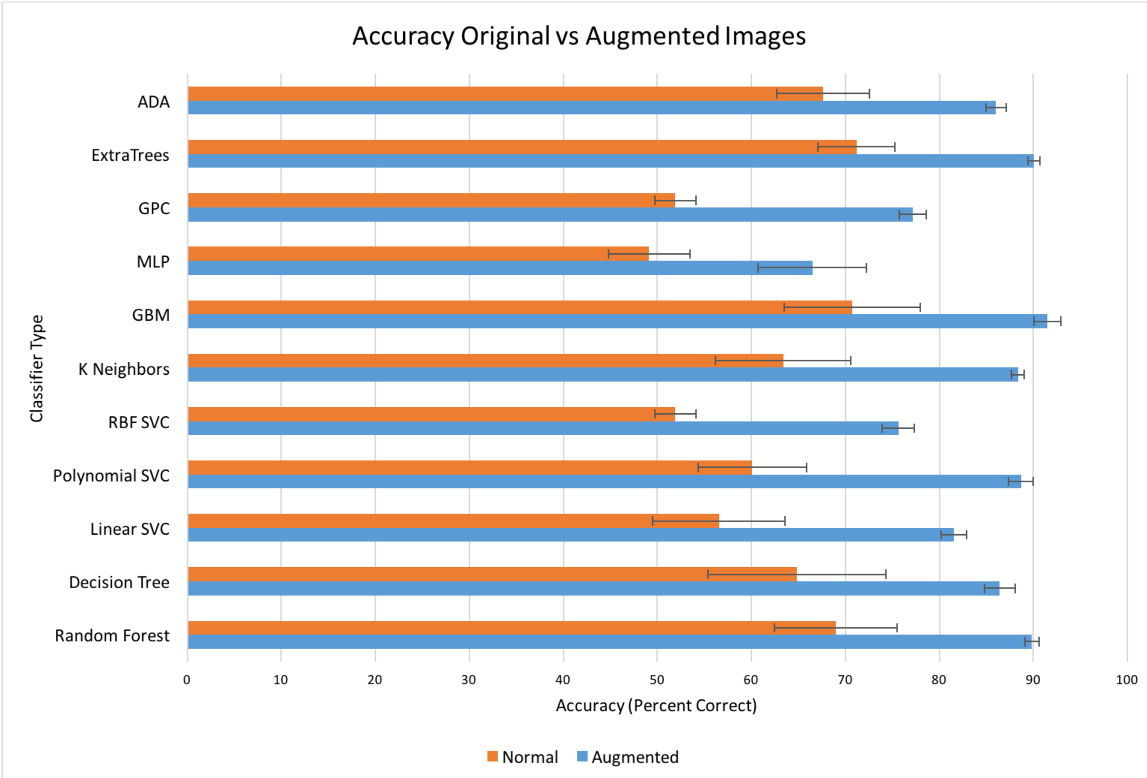
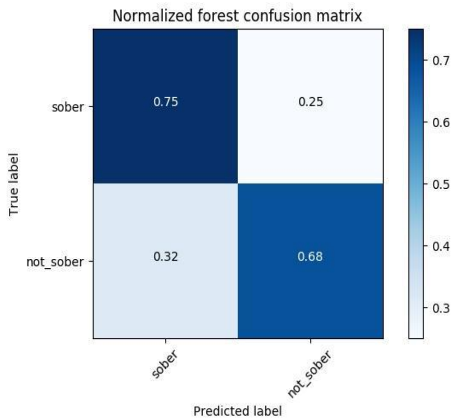



**Figure 31:** Accuracy of Augmented Images Sober or Not Sober.

The addition of augmented photos to our dataset did not seem to affect the breakdown of false or true positive rates (see Figures 32 and 33). Predicting sober remained more accurate than predicting not sober, and not sober people were slightly more likely to be labeled as sober, than a sober person being labeled as not sober.



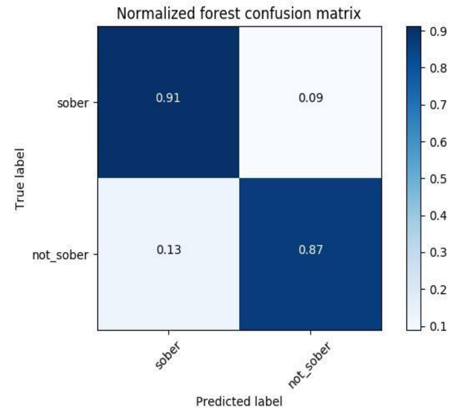




**Figure 32 (left)**: Normal Dataset Confusion Matrix

**Figure 33 (right)**: Augmented Dataset Confusion Matrix

## 3.6 Classifying Images into 4 Bins

We divided the *3 Glasses Later* dataset into 4 classes: Sober, 1 Glass, 2 Glasses, and 3 Glasses [28]. We then attempted to train machine learning classification models that could predict which class a given photograph belonged to. However, when classifying the images into four classes we struggled to achieve accurate results, therefore we utilized augmentation to help increase our training data sample set. Using a Random Forest, we achieved 36 percent accuracy in categorizing images, Table 3. This result was one of our best accuracies, with some models producing results worse than guessing (25 percent accuracy).

**Table 3:** Four Category Accuracy using Random Forest

| Dataset | Accuracy | SD |
|---|---|---|
| Normal | 36% | ±7% |
| Augmented | 88% | ±2% |

Trying to put images into the four categories is a very difficult task considering the difference between someone who is sober and had just one glass may be minimal. The confusion matrix for a Random Forest Classifier, as depicted in Figure 34, shows that the model often predicted someone who had consumed one drink as being completely sober.

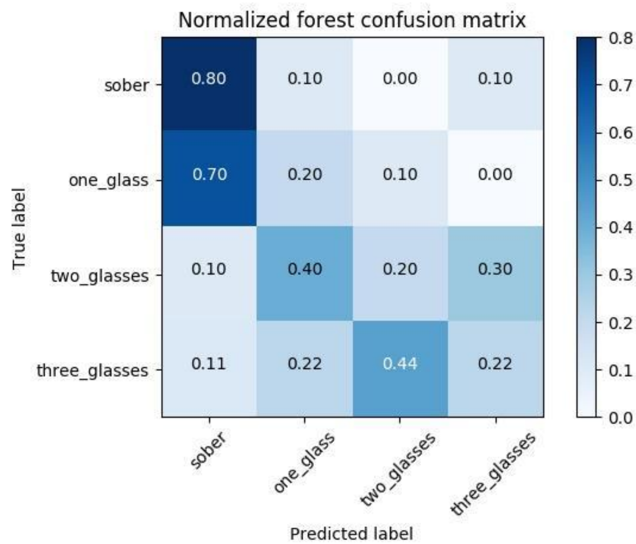



**Figure 34**: Confusion Matrix for the Random Forest classifier, Four Classes Confusion Matrix

In order to improve the accuracy of our models and evaluate the efficiency of our classification results in a real-world use case, we decided to divide the data into two categories Sober: no drinks or 1 drink, and Not Sober: 2 or 3 drinks (see Table 4). For most of our experiments we used these two classes to gauge accuracy since they reflect many countries' laws. Typically, people are allowed to drive after roughly one drink, but usually not after two drinks which is on the edge of the legal limit. After adjusting to this method we quickly observed results that were better than randomly guessing (50 percent accuracy), see Figure 35.

**Table 4**: Sober vs Not Sober Classes

| Sober | Not Sober |
|---|---|
| No Glasses | 2 Glasses |
| 1 Glass | 3 Glasses |

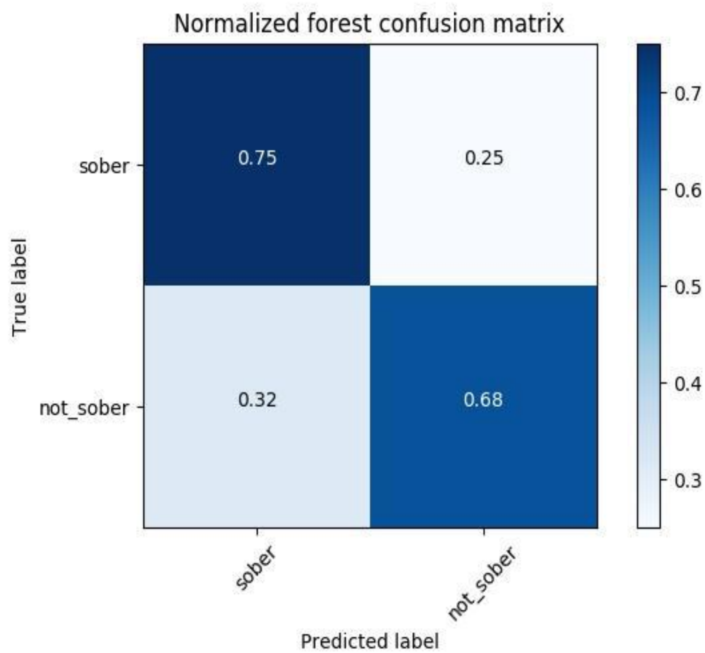



**Figure 35**: Confusion Matrix for the Random Forest Classifier, Two Classes Confusion Matrix

## 3.7 Removing Ineffective Features

To potentially improve accuracy, we reduced the number of features to use only the most effective features from our tests. We ran several classification models on different feature subset sizes, so we could determine if pruning insignificant features from the feature set would improve accuracy. In the charts below we display the performance changes when removing 0, 10, 50, 100, 150, 200, and 250 of the least important features when determining drunkenness. The feature set used in testing was a combination of landmark positions and landmark vectors totaling in about 268 features.

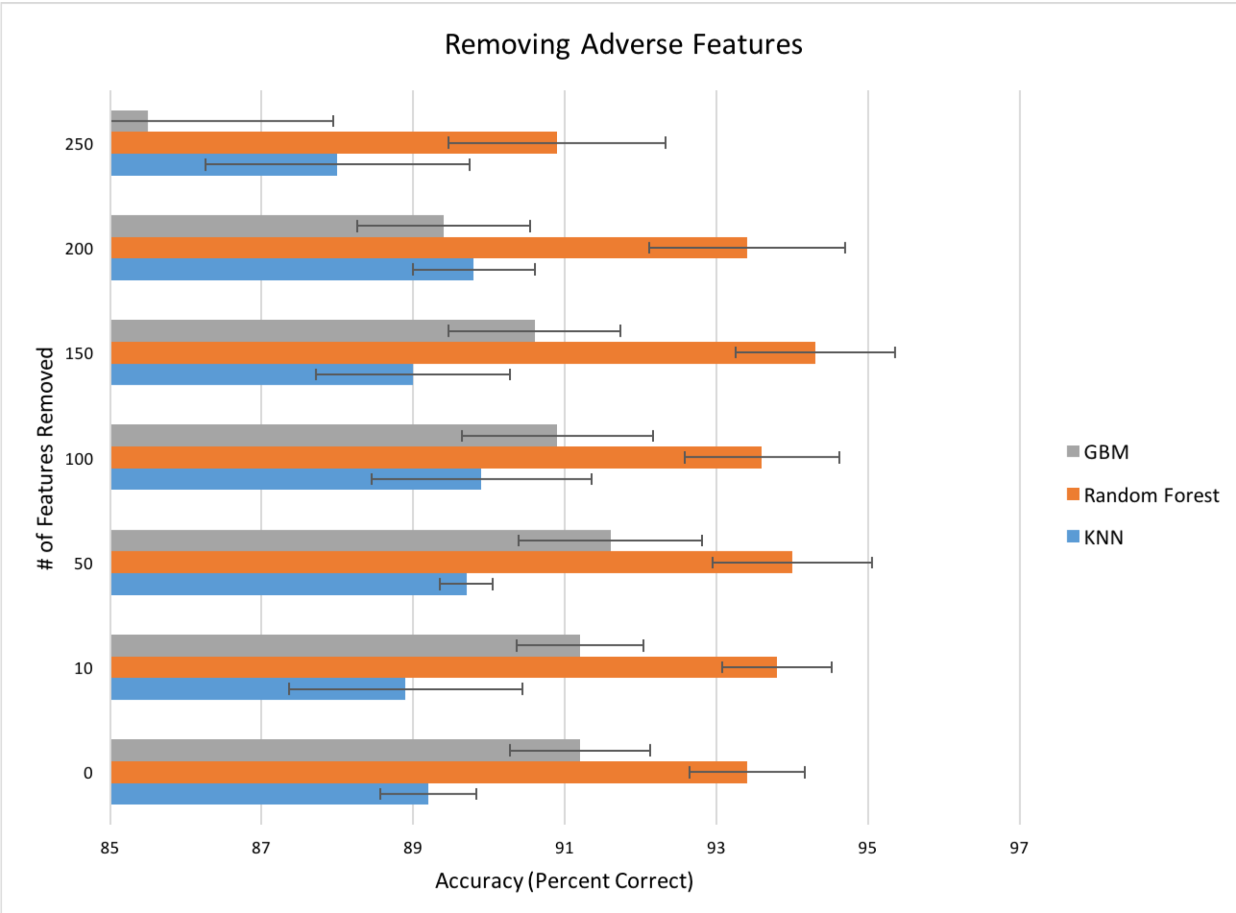



**Figure 36**: Number of Adverse Features Removed vs Accuracy

As shown in Figure 36 above, we found that removing 50 features helped improve the accuracy of determining if the subject was drunk or not. However, this varies between models and different folds and also was within the standard deviation of the accuracies of most of the models used. This led us to believe that it may not play an important role in making our models more accurate.

## 3.8 Trial Setup

Our code is setup to load a configuration file that specifies input features, dataset, classification models, number of folds, output format, removing of adverse features, and more. After every trial, this configuration is outputted to a file along with any results to a file so that the exact trial can easily be performed again if desired.

## 3.9 Cross-Validation

For each trial we cross-validated the data over 10 folds using an 80-20 split (train and test set) in our dataset. Specifically, a random selection of 80 percent of the images were used to train a classification model and then the remaining 20 percent of images were used to tested on the model in order to evaluate classification accuracy. This random 80-20 selection of images was then made for 10 trials (folds) each producing an accuracy from their testing set. We then computed the average accuracy across all folds for these tests to find a mean accuracy and percent error for a given classification model. Using cross-validation helped prevent our models from overfitting our data and ensured that our results generalized better. However, even with using cross validation, overfitting the training set is still possible.

## 3.10 Machine Learning Models

After building training and testing datasets, we used a variety of models to classify our results. The main models used were the Linear Support Vector Classifier, Polynomial SVC, Random Forest, and Decision Tree Classifier. Of these classifier types, Random Forest consistently performed the best. Figure 37, shows how the accuracy of the classifiers varied on standard features. Given that Random Forests generated consistently high scores and with lower variation between folds, our team decided to use Random Forests in our experiments to gauge how changing features impacted that accuracy.

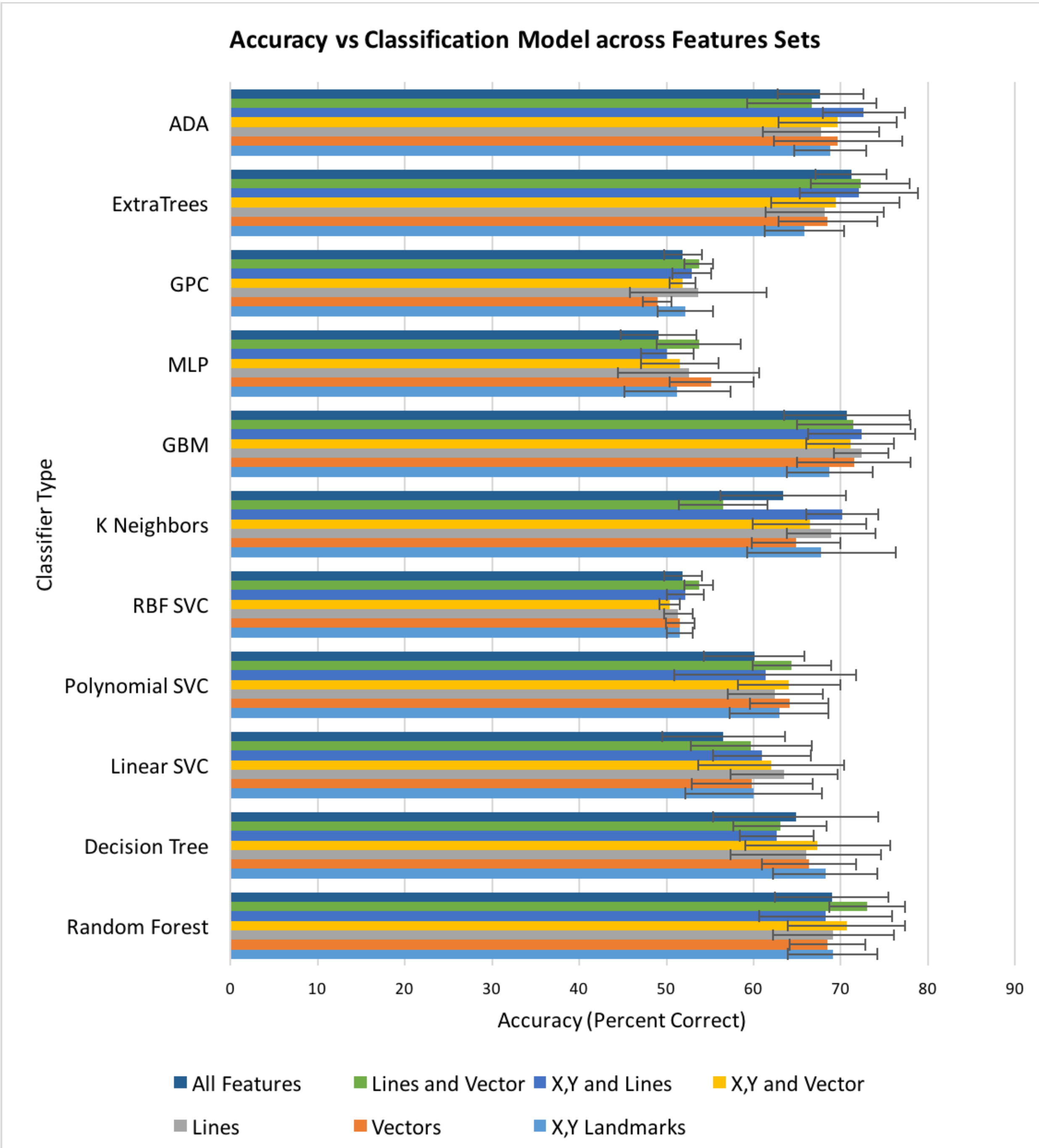



**Figure 37**: Classification Accuracies with various Features and Classifiers.

# 4 Analysis and Results

## 4.1 Combining Features

After identifying and extracting potential features for our classifier we tested various combinations of the feature sets. We experimented using facial landmark points, vectors, and lines. Figure 38 demonstrates that adding more features only slightly improved our accuracy. Using these combinations of features did reduce the error of our accuracies in tests. The use of multiple features sets combined with removing unimportant features increased the focus on important features in our classification models.

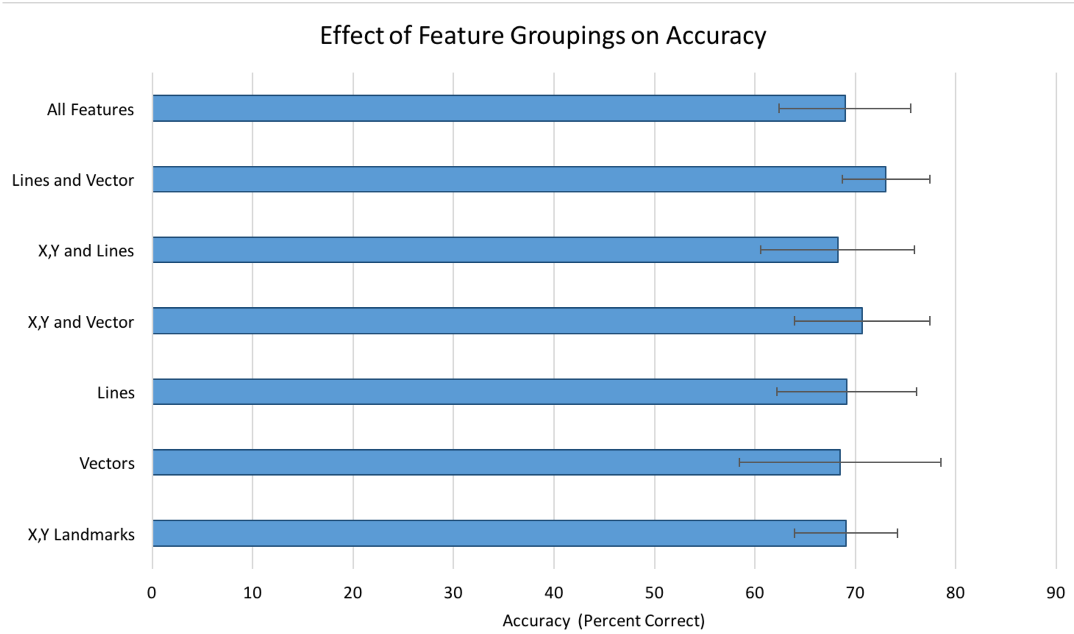



**Figure 38**: Accuracy of Random Forest with different sets of Features.

## 4.2 Summary of Results

In order to achieve are best accuracy at detecting drunkenness we used a combination of the features described above. Our model classified images from the original *3 Glasses Later* and the augmented dataset as either sober (0-1 drinks) or not sober (2-3 drinks). Due to concern that augmentations were too similar to their original photographs, we separated the photos such that people who appeared in the training set would not have any augmented photos placed in the testing set (and vice versa). This ensured the testing set did not leak in any way into the training set, and consequently dropped our accuracy from above 90 to 81 percent. To achieve 81 percent accuracy, we used the follow features:

- Landmark points - x,y (Section 3.4.1)
- Landmark vectors - length and direction (Section 3.4.2)
- Landmark lines - distances (Section 3.4.4)

We then normalized the images with the feature values from the subject's sober photograph by subtracting the sober values. Finally, we found that removing the 100 least important feature values from the set improved the accuracy the best. Using this combination of features in a Random Forest classifier, we were able to produce an accuracy of 81±3 percent at correctly classifying photographs as drunk or sober.

# 5 Prototype Android App

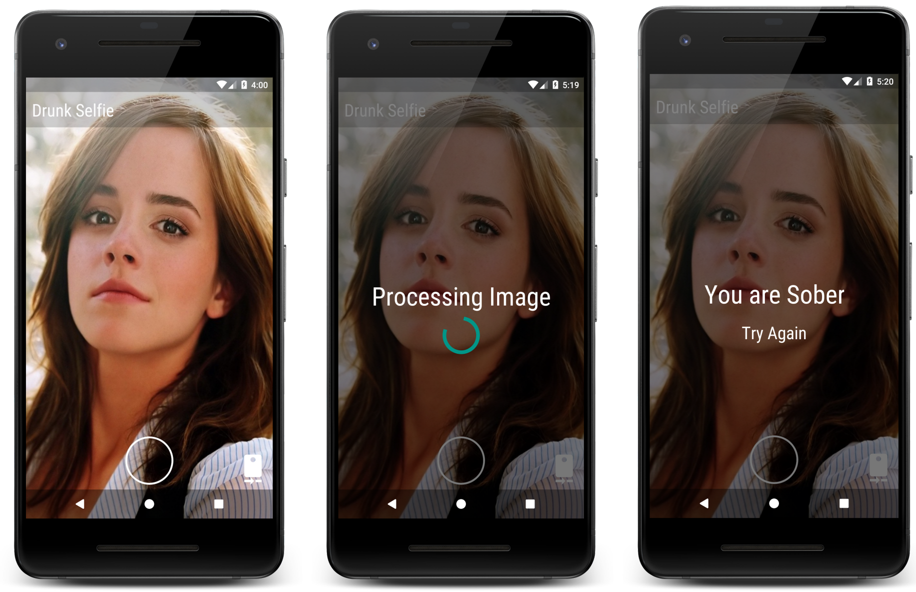



**Figure 39:** Prototype Android App

As a proof of concept for the application of our models, we created a prototype android app that is able to take photos and classify them using the models exported from our machine learning experiments. In order to improve the speed of the classification, and maintain a single python codebase for our classification, we used a remote linux server to process the images taken in the app, and send the results back to the phone after a classification is determined done. The app has a simple user interface, with a button for capturing a photo, and another for switching from front to rear camera.

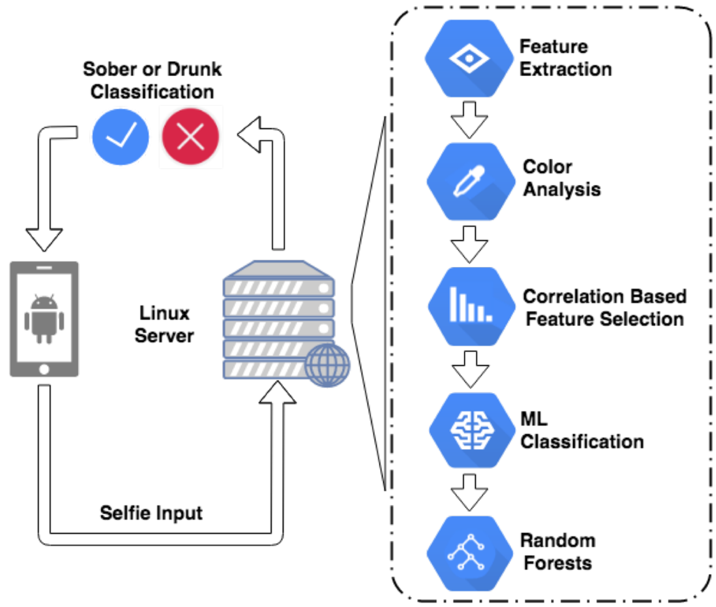



**Figure 40:** Prototype Drunk Selfie Android App

The standard android camera2 API is used to capture the photos. After a photo is captured, it's uploaded to our server for processing, and waits for a response. The Android AsyncTask and Java HTTP standard library is used for communicating with the server. An overview of the photo taking and classification process can be seen in Figure 40.

# 6 Conclusion

Impaired driving is a growing concern in the US and it accounts for “nearly one-third (29 percent) of the of all traffic-related deaths in the United States” [1]. In a world that is increasingly social and becoming more and more technological adept, the need for a mobile application to assist users in making smart decisions is growing. There are several applications on the market that attempt to help users track their drunkenness. However, they are cumbersome to use and require the user to answer numerous questions about their weight, height, etc. Our MQP addresses problem this by using a machine learning algorithm to determine the sobriety of the user from a selfie photograph, an engaging and simple way to use a phone.

Analyzing the 53 subjects in Marco Alberti’s *3 Glasses Later* dataset, we were able to identify what features most accurately indicate drunkenness. Furthermore, our team determined that using a combination of facial landmarks, vectors, and facial structures yielded the best results. Finally, we calculated that our best model produced an accuracy of around 81 percent at detecting when a subject from Marco Alberti’s photograph collection was either sober or drunk (binary classifier).

# 7 Limitations and Future Work

Our team believes that our model and android prototype can be further improved and expanded upon with new features and capabilities. Unfortunately, we felt that certain key elements of the project limited the possibility for success for our application.

1. Limited Dataset: The dataset provided by Marco Alberti's unfortunately limited the capability of the app and that it meant that we used photos that were staged and not real world representations of drinking atmospheres [28]. Augmentation of the original photos helped but it had its limitations and unfortunately using social media images did not allow us to have an accurate understanding of the subject's BAC. We could not use the social media posts for precise testing since we did not have the accurate drink number of the subject. However, with improvements to the dataset, such as more subjects, a less staged and more real world environment for the photos, and including Blood Alcohol Content (BAC) readings in each photo we would be able to improve the accuracy of our model. With the use of a more comprehensive dataset we also believe that vast improvements could be made to accurately labeling the user as having drank x number of drinks.
2. Simplified Classification: Currently our system does not accurately place each subject into the categories of drinking 0, 1, 2, or 3 glasses of wine. However, if we had access to a dataset with more subjects and records of their BACs then we could generate a model that more accurately placed each subject into their corresponding drink number category.
3. Increased Smartphone Computing Capacity: With improvements to smartphone technology we believe that including the feature extraction and classification on the mobile device would be a large improvement. Having the smartphone run all of the photo analysis processes would allow us to remove the assumption that the user has internet access. This would make the smartphone app more usable and less prone to failure if internet connection is lost.

Nevertheless, even without these improvements we believe that the system we implemented would still assist people in making more responsible decisions.

# References

[1] CDC Injury Center, "Impaired Driving: Get the Facts," *CDC Injury Center*, 2017. [Online]. Available: https://www.cdc.gov/motorvehiclesafety/impaired_driving/impaired-drv_factsheet.html. [Accessed Oct. 12, 2017].

[2] C. Ingraham, "Think you drink a lot? This chart will tell you," *Washington Post*, Sept. 25, 2014. [Online]. Available: https://www.washingtonpost.com/news/wonk/wp/2014/09/25/think-you-drink-a-lot-this-chart-will-tell-you/. [Accessed Oct. 12, 2017].

[3] U.S.C. Bureau, "2014 ACS 1-year PUMS," *U.S.C. Bureau*, 2014. [Online]. Available: http://factfinder.census.gov/. [Accessed Oct. 12, 2017].

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

[5] 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.

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

[7] G. Bergen, J. Yao, R.A. Shults, E. Romano and J.H. Lacey, "Characteristics of designated drivers and their passengers from the 2007 National Roadside Survey in the United States," *Traffic Injury Prevention*, vol. 15, pp. 273-277, 2014.

[8] Anonymous "Mobile Fact Sheet," *Pew Research Center*. Jan 12, 2017. [Online]. Available: http://www.pewinternet.org/fact-sheet/mobile/. [Accessed Sept. 20, 2017].

[9] 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.

[10] Statista, "Share of adults in the United States who have ever taken a selfie as of February 2017, by age group," *Statista*, February 27, 2017. [Online]. Available:

https://www.statista.com/statistics/304861/us-adults-shared-selfie-generation/. [Accessed Oct. 12, 2017].

[11] University of Notre Dame, "Blood Alcohol Concentration," *University of Notre Dame*, 2017. [Online]. Available: https://mcwell.nd.edu/your-well-being/physical-well-being/alcohol/blood-alcohol-concentration/. [Accessed Oct. 12, 2017].

[12] BACtrack, "Three Types of BAC Testing," *BACtrack, 2017*. [Online]. Available: https://www.bactrack.com/blogs/expert-center/35043461-three-types-of-bac-testing/. [Accessed Oct. 12, 2017].

[13] DUI California, 2017 Available: "http://www.dui-california.com/graphics/810fml11.jpg,”.

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

[15] C. You, N. Lane, F. Chen, R. Wang, Z. Chen, T. Bao, M. Montes-de-Oca, Y. Cheng, M. Lin, L. Torresani and A. Campbell, "CarSafe app," *MobiSys*, pp. 13-26, Jun 25, 2013.

[16] 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.

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

[18] V. Neagoe and S.Carata, "Subject Independent Drunkenness Detection Using Pulse-Coupled Neural Network Segmentation of Thermal Infrared Facial Imagery," *Mathematical and Computational Methods*, vol. 1, pp. 305-312 2016.

[19] L.S. Moulin, A.A. Da Silva, M.A. El-Sharkawi and R.J. Marks, "Support vector machines for transient stability analysis of large-scale power systems," *IEEE Trans. Power Syst.*, vol. 19, pp. 818-825, 2004.

[20] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, vol. 3.4.1, 2000.

[21] Pedregosa, "Scikit-learn: Machine Learning in Python," *JMLR* vol. 12, 2011.

[22] D.W. Ruck, S.K. Rogers and M. Kabrisky, "Feature selection using a multilayer perceptron," *J.Neural Network Comput.*, vol. 2, pp. 40-48, 1990.

[23] L. Breiman, "Random forests," *Mach.Learning*, vol. 45, pp. 5-32, 2001.

[24] Hanselmann, Michael & Köthe, Ullrich & Kirchner, Marc & Y Renard, Bernhard & Amstalden van Hove, Erika & Glunde, Kristine & Heeren, Ron & A Hamprecht, Fred & H Morgan, Russell, "Towards Digital Staining using Imaging Mass Spectrometry and Random Forests-Technical Report," *ResearchGate*, 2009.

[25] A. Sabharwal, "Google Photos: One year, 200 million users, and a whole lot of selfies," *Google Blog.* May 27, 2016. [Online]. Available: https://blog.google/products/photos/google-photos-one-year-200-million/. [Accessed Oct. 12, 2017].

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

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

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

[29] Rein-Lien Hsu, M. Abdel-Mottaleb and A. K. Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 696-706, 2002.

[30] B. Başa, "Implementation of Hog Edge Detection Algorithm On fpga's," *Procedia - Social and Behavioral Sciences*, vol. 174, pp. 1567-1575, 2015.

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

[32] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867-1874, 2014.

[33] P. van Gent, "Emotion Recognition With Python, OpenCV and a Face Dataset," , April 01, 2016. [Online]. Available: http://www.paulvangent.com/2016/04/01/emotion-recognition-with-python-opencv-and-a-face-dataset/. [Accessed Feb. 20, 2018].

[34] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986. DOI: 10.1109/TPAMI.1986.4767851

[35] A. Jung, "imgaug," vol. 0.2.5, 2017. [Online]. Available: http://imgaug.readthedocs.io/en/latest/index.html/. [Accessed Feb. 20, 2018].