Unsafe driving Detection
Classification of driving activities into safe and unsafe

Problem Statement

Our project aims at detecting distracted activities while driving by using images taken from dashboard camera of the car. Our goal is to classify images of activities during driving as either safe driving or distracted driving ("using cellphone" or "doing make-up or hair").

Application Motivation

As defined by US Government, *"Distracted driving is any activity that could divert a person's attention away from the primary task of driving. All distractions endanger driver, passenger, and bystander safety". [1]* Distracted driving activities are alarmingly increasing all over the world. In United States alone the statistics report that, "*In 2014, 3,179 people were killed, and 431,000 were injured in motor vehicle crashes involving distracted drivers"[1].*

We aim to ease this alarming statistics by building a system that can generate alerts for distracted driving.

Executive Summary

Data source

We have taken the data from State Farm's Distracted Driving Data hosted on Kaggle[2]. This data contains labelled examples of the images of drivers doing various activities while driving such as using cellphone, doing make-up and hair or driving safely.

Classification of Driving into Safe and Unsafe Driving

Our main goal is classifying images into safe vs unsafe. However, it is also important to distinguish the type of distraction during driving. For example, it is important to distinguish between talking on cellphone while driving or doing hair and makeup while driving rather than just classifying both of these as unsafe. Although, both doing make-up and talking using mobile phone can lead to distraction during driving, the law related to both these activities are different. Moreover, the amount of distraction created due to both of these activities are also different. So, we are classifying driving images into three categories namely, safe driving, talking using cellphone or doing make-up or hair.
The example images for these categories are as shown below:

Safe driving

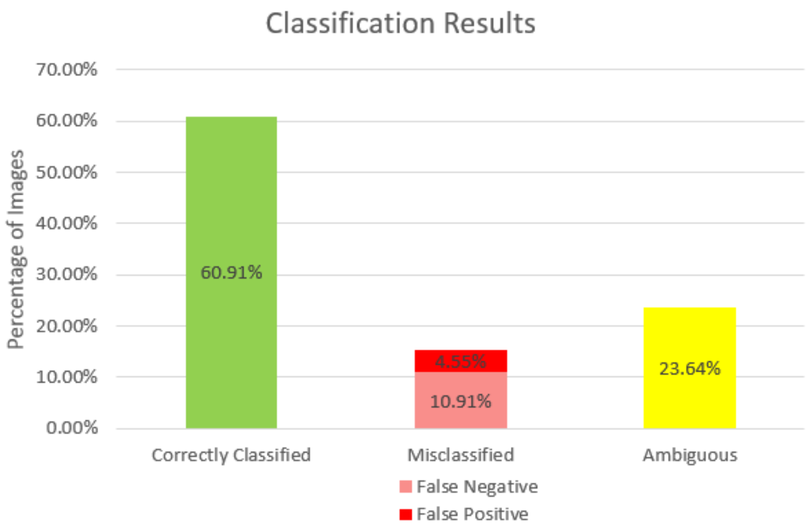
Using cellphone

Applying Make-up

Successful classification

We are successfully able to classify images into one of the three classes described above with an accuracy of about 61% which is an improvement of about 100% over the random classifier.

The following bar graph shows how the classification model classifies the images. The classifier is designed such that it will avoid crisp classification of ambiguous images. It defines a separate category to classify ambiguous images, which is sent for manual classification. In our case we have about 25% images as ambiguous.



For example, the image shown here are classified as safe and unsafe (doing hair or makeUp respectively)

Safe driving Doing hair or make-up

However, the following images is classified as ambiguous and is passed for manual classification

Actual: Safe
Classified: Ambiguous

Actual: Calling
Classified: Ambiguous

Implications of misclassification

There are several implications of misclassification for our application. If we classify unsafe driving as safe, it can lead to serious losses like accidents. So, our aim is to keep the false positive rate as low as possible, as the cost of this misclassification is very high.

At the same time, if our model ends up raising too many false alerts, it may lead to the users of the system becoming averse to alerts of distracted driving. So, in order to reduce the rate of accidents as well as false alarms, we avoid classifying images which are ambiguous and cannot be classified into one of the four classes with confidence. These images are passed on for manual classification.

Technical Approach

In order to classify images into three classes mentioned above, we went through several steps which were a congregation of standard image processing steps and various application specific designs and algorithms.

Step 1 : Understanding data and Data analysis

Data consists of different people who are simulating various safe or distracted driving activities in motion. The data represents various still images captured during the motion. So, training data has many similar images of certain people doing the same activity, with a slight difference of movement in the picture. To avoid bias in the classification due to similar images, we are using a completely different set of images of different people for testing rather than doing cross validation with the training data to measure the performance of our model.

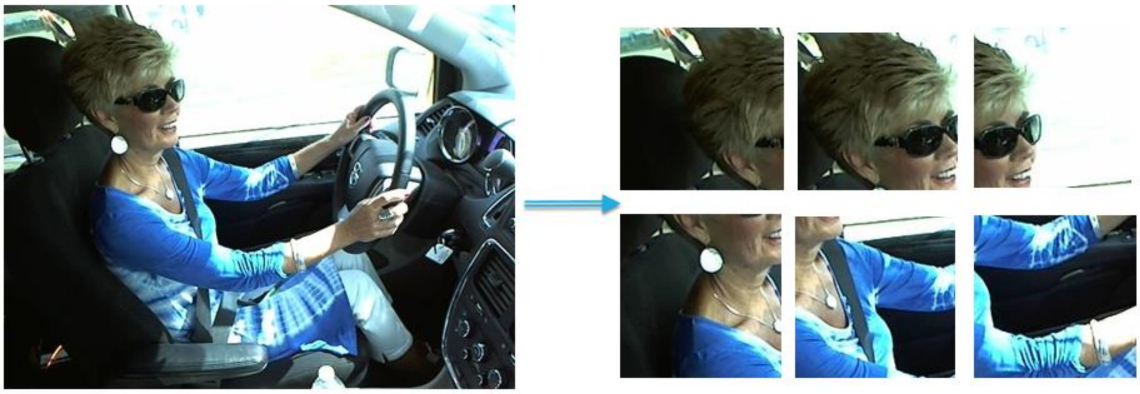
Step 2: Face Detection

Problem Statement: Since, most of the distracted driving activities like talking using cell-phone and doing make-up and hair while driving are detectable through facial gestures, we decided that we should input only faces to our classification model. This helps us eliminate the noise being passed to the classifier.

Problems faced with standard face detection: We started out by experimenting with standard face detection algorithms. However, the challenge was these algorithms were not very well suited for our applications. For example, Viola–Jones object detection framework is one of the most extensive object detection framework that has training examples from standard objects. While Viola Jones has a very exhaustive face detection package, most of its training examples is frontal face images as the algorithm is designed to recognize Haar like features such as eye, nose, mouth etc. [3] As most of the images we have are side faces and faces are many times covered partially or almost completely like the image shown below, Viola Jones did not work well on our dataset.

Images that are partially covered with hair, goggles or cell phone

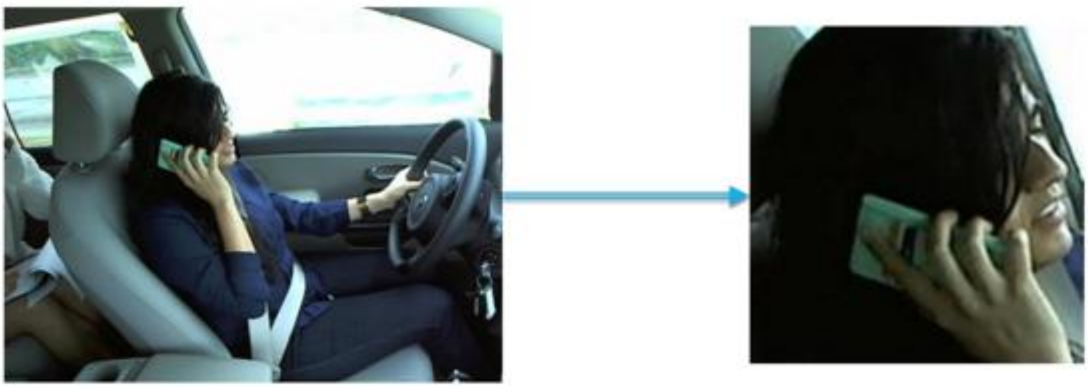
Our solution to the challenge of face detection: We implemented a simple version of face detection algorithm inspired by Viola Jones, itself. We created a window of 200 x 200 pixels and slide that window through the upper left 25% of the image. Given the relative positioning of the dashboard camera of the car to the car seat and steering, we are sure that top left quarter captures the face almost all the time. Then, we labelled the windows of 150 such images as either face or non face, across three classes of images.
For example, in the following image a sliding window capture of top left corner yields following set of images.



1 Original Image   2 Sliding window partial images

Using, this manually labelled data of sliding windows of 150 images, we trained a data model consisting of HOG (Histogram of Gradients) + SVM (Support Vector Machines) to detect face in the images.

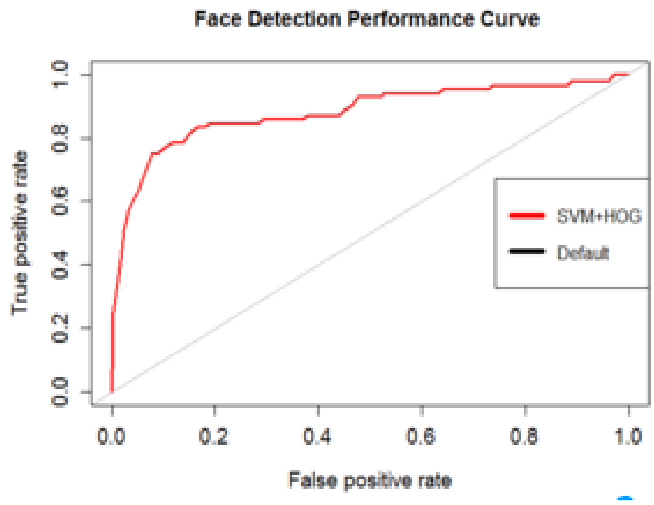
Output of face detection: Our face detection algorithm was able to recognize faces with an accuracy of 79.8%. We were able to classify images which were almost or partially covered like the image shown below.



Successful face detection even though face almost covered

On investigation, we found that our HOG algorithm was picking up the features such as the boundary of the driving seat , which is always placed just behind the face. This detection enhances the classifiers capability to identify faces even though they are side faced or partially covered.

The following was the ROC curve and confusion matrix of our face detection algorithm.



	Predicted Face	Predicted Not-Face
Actual Face	280	5
Actual Not-Face	400	1315

This face detection model was used on 2000 images, 500 of each class, namely safe driving, using cellphone on left, using cellphone on right and doing make-up and hair.

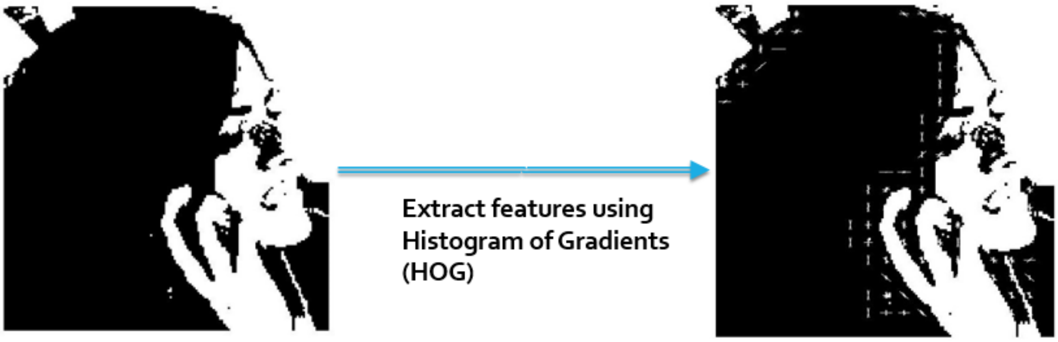
Step 3: Image preprocessing

Binarizing the image: On research in the domain of computer vision, we found that certain measures of pre-processing are essential for building a successful classification model. In order to eliminate the noise such as variation in color intensity, brightness, skin tone, clothing color etc, we convert the images into binarized images as follows:

Using Histogram of Gradients (HOG) for feature selection

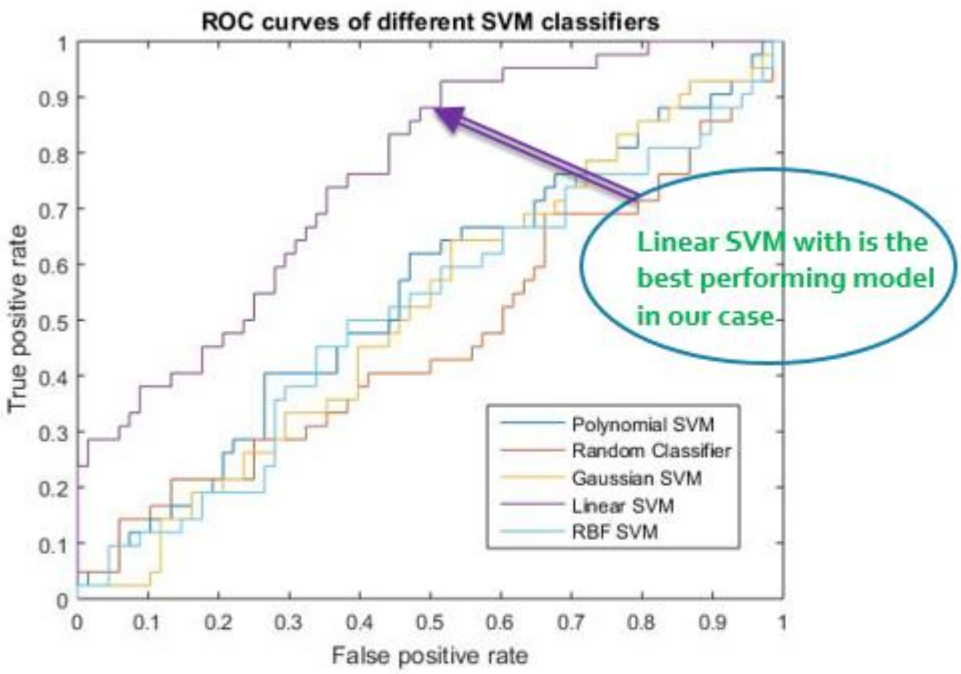
We pinned down our basic task of classification into identifying the presence of certain objects or hand on face. For example presence of a mobile phone, lipstick or the hand on face while doing certain distracted activity like adjusting hair and talking on phone would let us know the distracted driving activity. So, HOG (Histogram of Gradients) algorithm which is a feature descriptor used in computer vision and image processing for the purpose of object detection, works best in our case.[4]

We achieved the following type of featurization result after using HOG, where HOG is able to identify the edges of the objects in the image. This output is further passed to classifier such as SVM which identifies the object in the image



Step 4: Classification using Support Vector Machines (SVM):
It is said about SVM classifier that “Once trained on images containing some particular object, the SVM classifier can make decisions regarding the presence of an object, in additional test images” [5]. This property of the SVM classifier to identify objects based on training via supervised learning works best on the output of HOG. So, we try various models of SVM with varying kernel functions like linear, polynomial, Gaussian and Radial Basis Function (RBF).
The ROC curve of each of these functions in shown in the graph below. Here, linear SVM seems to work best for our application.

Please note that the ROC curves below reflect the curve of classification for safe vs other classes which include make-up and using cell phone. So, this is a multi-class output converted into two class output. Random classifier would have an accuracy of 33% in this case.



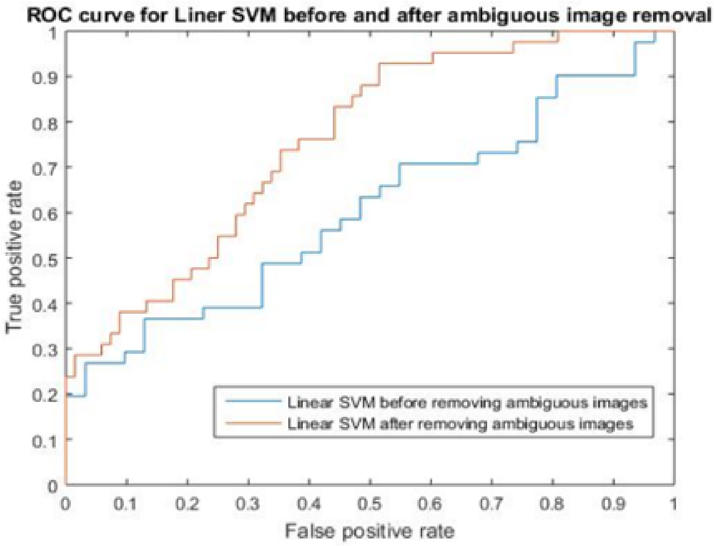
Step 5: Optimization and selecting threshold
Once we choose our model as HOG + Linear SVM, we look at the optimization measures for improving the model for HOG + Linear SVM. Our goal of optimization in this application is to decrease the rate of accidents by avoiding false positives. As the cost of undetected distraction in driving is much more than cost of false alarms raised by false negatives, we need to set our

threshold on the leftmost region of ROC curve which leads us to a threshold of 0.76 to classify between safe and unsafe classes

However, we do not want to increase the false alarm rate so much that it will lead to averseness of the law officers in detecting distracted driving.

So in order to have a solution that will lead to decrease in misclassification, we avoid a crisp classification of images that lie on the boundary of the classification. We decide to classify the images that lie within the -5% and +5% of the threshold value as ambiguous. We pass on this images for manual classification. This approach will not only improve our accuracy of classification but will lead to users of the system manually classifying only ambiguous images rather than having to look at all false negatives.

After removing ambiguous images, we get the following accuracy and new ROC curve which is about 8% improvement from the previous classification.



	Predicted Safe	Predicted Calling	Predicted Make-Up
Actual Safe	42	6	39
Actual Calling	9	81	3
Actual MakeUp	9	3	78

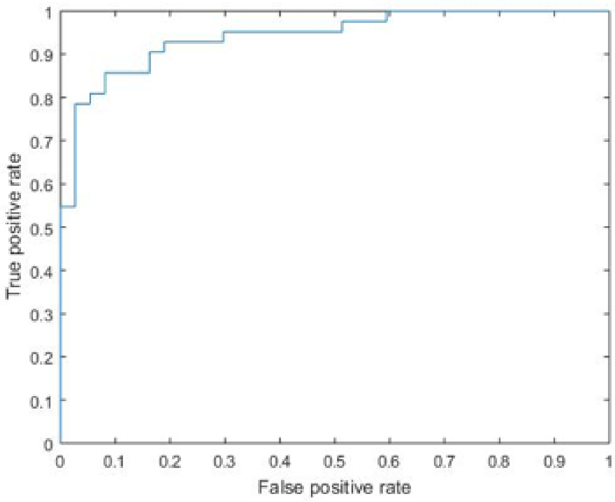
Confusion matrix of linear SVM after eliminating ambiguous images

Analysis of Output

We observe that our classification of images into three classes, namely calling, safe driving and doing hair and makeup is performing 2 times better than random classifier. It gives us an accuracy of 61% as opposed to random classifier accuracy of 33% for 3 class output. Our accuracy further improves significantly by declining to classify ambiguous images crisply in one class or other. We send this images for manual classification.

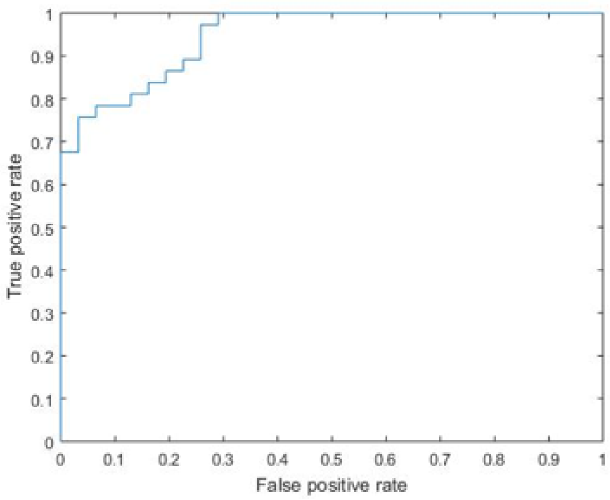
We further analyze the output of each of the two classes among the three classes, to observe which classes are most easy of difficult to classify for our model.

Our analysis of output shows that the difference between class safe driving and calling using cell phone is perfectly distinguishable with an accuracy of 94.27%, as seen in the following 2 class ROC curve.



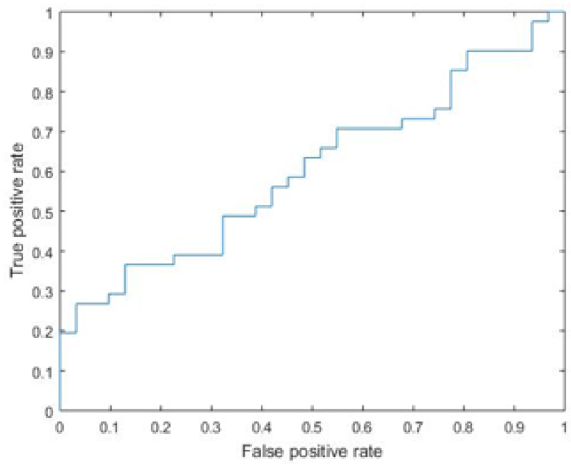
	Predicted Safe	Predicted Calling
Actual Safe	108	18
Actual Calling	12	96

Similarly, our classifier is able to distinguish between safe driving and doing hair and makeup while driving with an accuracy 84.62%



	Predicted Safe	Predicted Hair & Make-up
Actual Safe	93	15
Actual Hair & Make-up	15	72

But, the classification among class Hair and make-up and calling using cell phone is more difficult. Our 2 class model performs with an accuracy of 60.35% for this classification.



	Predicted Calling	Predicted Hair and Make-up
Actual Calling	45	75
Actual Hair & Make-up	15	63

This goes by our intuition that since the features of class calling using cell phone and doing hair and makeup are similar in terms of hand gestures that are covering the face, so there is likely to be ambiguity in their classification.

Ambiguous Hair and MakeUp Ambiguous Calling

Future Work

While doing this project we realised that there are several ways in which this project can further explored in a meaningful way. We mention some of the thoughts for further exploration of this project as below:

Reinforcement Learning

Reinforcement Learning is a field of study that focusses on improving the model based on a notion of cumulative reward. In our application, we let users classify ambiguous images manually. We can make use of this manual classification output in improving our model through reinforcement learning.

Moreover, in different states and countries, notion of safe vs unsafe and legal vs illegal may be different. With the help of reinforcement learning, we can start with a common model for all states and countries and based on manual classification labels from users for ambiguous images, we can build country or state specific models for distracted driving detection.

Pre-trained neural networks
There are several pre-trained convolutional neural networks out in the market that can detect side faces and extract important features of human faces. By the use of such pre-trained networks, we will not need to do feature selection and other pre-processing to the image. We can further modify these pre-trained models to adapt to our specific application.

References:

1. http://www.distraction.gov/stats-research-laws/facts-and-statistics.html
2. https://www.kaggle.com/c/state-farm-distracted-driver-detection/data
3. https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework
4. https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
5. https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients