Driver Cell Phone Usage Detection on Strategic Highway Research Program (SHRP2) Face View Videos

Keshav Seshadri[1], Felix Juefei-Xu[1], Dipan K. Pal[1], Marios Savvides[1] and Craig P. Thor[2] *

Abstract

The harmful effects of cell phone usage on driver behavior have been well investigated and the growing problem has motivated several several research efforts aimed at developing automated cell phone usage detection systems. Computer vision based approaches for dealing with this problem have only emerged in recent years. In this paper, we present a vision based method to automatically determine if a driver is holding a cell phone close to one of his/her ears (thus keeping only one hand on the steering wheel) and quantitatively demonstrate the method's efficacy on challenging Strategic Highway Research Program (SHRP2) face view videos from the head pose validation data that was acquired to monitor driver head pose variation under naturalistic driving conditions. To the best of our knowledge, this is the first such evaluation carried out using this relatively new data. Our approach utilizes the Supervised Descent Method (SDM) based facial landmark tracking algorithm to track the locations of facial landmarks in order to extract a crop of the region of interest. Following this, features are extracted from the crop and are classified using previously trained classifiers in order to determine if a driver is holding a cell phone. We adopt a through approach and benchmark the performance obtained using raw pixels and Histogram of Oriented Gradients (HOG) features in combination with various classifiers.

1. Introduction

The number of deaths due to distractions caused by cell phone usage during driving are on the rise, not just in the US but across the world. In 2013, $3,154$ people lost their lives and an estimated $424,000$ were injured in the US due to a distracted driver [1]. Distraction due to cell phone usage constitutes a sizable portion of the statistic with 18% of the incidents involving cell phone usage in 2009. Studies in a simulated driving environment under controlled settings have shown that impairment associated with using a cell phone while driving can be as profound as those associated with driving while drunk [23]. Braking reactions were delayed when drivers were conversing on a cell phone, leading to more traffic accidents [22, 23]. Therefore, it is becoming increasingly important to accurately detect cell phone usage by drivers, both from the safety and law enforcement points of view.

In order to study the more general problem of driver behavior, the Federal Highway Administration (FHWA) recently commissioned an exploratory project that challenged researchers in university and industry to develop computer vision and machine learning based algorithms that were capable of processing naturalistic videos of drivers and detecting signs of tiredness in drivers, cell phone usage by drivers, tracking head pose, monitoring if the driver had both hands on the steering wheel, *etc*. The driver monitoring algorithms that can be developed will be useful for two reasons. Firstly, they could potentially be deployed in a real-world scenario for driver monitoring as part of a law enforcement effort. Secondly, they could be used to automate the process of annotating the videos that have been already collected or collected for a future study. Currently, videos collected have to be manually annotated and processed (sometimes on a frame by frame basis) in order to provide researchers with ground truth data, which in turn limits the scale of such data sets. This in turn limits the scale of the studies. It is in this context that our work aims at addressing the specific problem of detecting whether a driver is holding a cell phone in one hand and using only one hand to control the steering wheel of a vehicle. Our results indicate that a fairly high level of accuracy, that is competitive with state-of-the-art results obtained on the same problem, can be obtained using minimal training data.

The rest of this paper is organized as follows. Section 2 reviews some of the prior work carried out on vision and non-vision based driver monitoring systems and algorithms. Section 3 provides details on the data that we used in our study and delves into the methodology that we employed in

*

[1] The authors are with the CyLab Biometrics Center and the Department of Electrical and Computer Engineering (ECE), Carnegie Mellon University, Pittsburgh, USA. e-mail: kseshadr@andrew.cmu.edu, felixu@cmu.edu, dipanp@andrew.cmu.edu, and marioss@andrew.cmu.edu
[2] The author is with the Office of Safety Research and Development, Federal Highway Administration, U.S. Department of Transportation, Virginia, USA. email: Craig.Thor@dot.gov

order to determine if a driver is using a cell phone, *i.e.*, holding it up to his/her ear and thus keeping only one hand on the steering wheel of a car, or not. Section 4 details the experimental protocols that we followed and the results that were obtained, and finally, section 5 presents some concluding remarks and highlights some possible research directions to pursue in future work.

2. Related Work

There has been a lot of recent work in the broad area of driver behavior monitoring and the specific problem of driver cell phone usage detection. Artan *et al.* [4] used data captured by a highway transportation imaging system, which was installed to manage High Occupancy Vehicle (HOV) and High Occupancy Tolling (HOT) lanes, for detecting cell phone usage by drivers. The cameras used were situated at an elevated position pointing towards the approaching traffic with Near Infrared (NIR) capability to tackle night vision. After the images were acquired, the authors adopted a series of computer vision and machine learning techniques for detection and classification. They first used a Deformable Part Model (DPM) [14] to localize the windshield region within the image and used a DPM based simultaneous face detection, pose estimation, and landmark localization algorithm developed by Zhu and Ramanan [30] to locate the facial region and crop out a region of interest around the face to check for the presence of a cell phone. Finally, image descriptors extracted from the crops were aggregated to produce a vector representation which was classified using a Support Vector Machine (SVM) [11] classifier to determine if the driver was using a cell phone or not.

Zhang *et al.* [29] also studied a similar problem. In their work however, the camera acquiring the video footage was mounted above the dashboard of a car. They extracted features from the face, mouth, and hand regions and then passed them passed on to a Hidden Conditional Random Fields (HCRF) model for final cell phone usage classification. For face detection, they used a cascaded AdaBoost [15] classifier with Haar-like features [24]. For mouth detection, a simple color-based approach was found to be sufficient because the red component in the mouth region is stronger than the rest of facial region, and the blue component is weaker. Therefore, they operated in the YC_bC_r color space and measured the ratio of C_r/C_b as their cue for mouth region detection. For the detecting hand region, they incorporated both color and motion information.

There has also been some recent research on non-vision based approaches for detecting cell phone usage by drivers. Bo *et al.* [6] leveraged various sensors integrated in today's smartphones, such as accelerometers, gyroscopes, and magnetometer sensors, to distinguish between whether a phone was being used by a driver or a passenger. Yang *et al.* [28] harnessed a car's stereo system and Bluetooth network in an acoustic based approach to estimate the distance of a cell phone in use from the car's center and were thus able to determine whether the user was the driver or not. Breed *et al.* [7] monitored emissions from a cell phone by placing three directional antennas at various locations inside a car. A receiver was associated with each antenna and included an amplifier and a rectifier module that converted radio frequency signals to DC signals which were used to tell which antenna provided the strongest signal. A correlation could then be made for finding the most likely location of a cell phone being used by an occupant in the car.

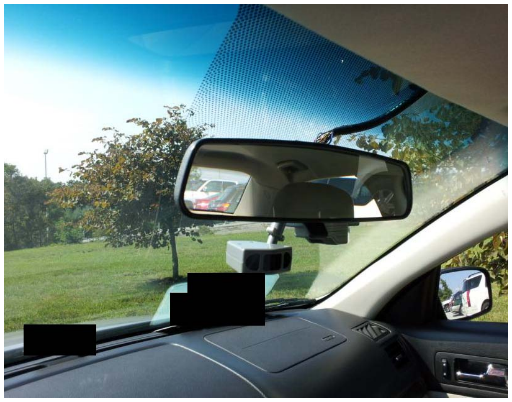

[redacted]
[redacted]

Figure 1: The setup of the DAS head unit and cameras that was used for acquisition of the mask head pose validation data. This image has been reproduced (with some modifications) from a document providing an overview of the mask head pose validation data that was obtained after signing a data sharing agreement. Certain portions of the image have been covered with black patches in order to prevent the dissemination of any information that is not to be made public under the terms of the data sharing agreement.

3. Our Approach for Cell Phone Usage Detection

This section provides details on our approach for automated cell phone usage detection. Our approach was designed for use on data that was acquired for a study of naturalistic driving behavior and in order to better understand our approach, it is first necessary to provide details on the setup used to acquire the data and data itself.

3.1. Details on the Data Used in Our Study

For carrying out our research, we used naturalistic driver behavior data that was acquired to evaluate the capability of the Virginia Tech Transportation Institute (VTTI) [3] head pose estimation system, referred to as mask. The platform for collecting the data was a 2001 Saab $9-3$ equipped with two proprietary Data Acquisition Systems (DAS). The collected data included digital video, GPS position and heading, acceleration, rotation rates, and ambient lighting collected at rate that varied from varied from 1Hz to 15Hz. The DAS units also collected data produced by the mask system. The participant was seated in driver's seat of the car and an experimenter (equipped with a laptop) was present with the participant. The experimenter supervised data collection and provided guidance to the participant. A handheld trigger connected to one of the DAS units allowed the experimenter to annotate the DAS data stream whenever an event of interest occurred. In order to collect the participant's face view videos, a camera was mounted below the rear view mirror, as shown in Figure 1.

One of the DAS units collected a single channel of minimally compressed (resolution of 720×480), full face digital video at 15 frames per second. The other DAS unit collected standard Strategic Highway Research Program (SHRP2) [2] videos. The two video streams were aligned using GPS timestamps that were recorded. The SHRP2 video comprised of four channels of video, forward view, face view (resolution of 356×240), lap and hand view, and rearward view, recorded at 15 frames per second and compressed into a single quad video, as shown in Figure 2. It is the SHRP2 face view videos that we use in this work.

Some of the SHRP2 videos were acquired when the participant was seated in a static vehicle, while others were acquired when the participant was driving. The environmental conditions (time of day) also varied in the videos. In the static vehicle trials, the data was acquired in a research lot at VTTI with each of the 24 participants asked to perform a series of glances to predefined locations (such as the left window or mirror, forward windshield, center console, *etc*.) or to simulate a brief cell phone conversation. Each static vehicle participant was asked to wear four pairs of eyeglasses (including a pair of sunglasses) and a baseball cap and complete the glancing and cell phone simulation tasks under these varying cases. The dynamic vehicle trials were conducted on a predefined route (approximately 15 miles long including a variety of road types) around Blacksburg, Virginia. Over the course of the drive, each of the 24 participants were asked to perform various tasks that included reporting the vehicle's speed, turning the radio on and off, locating a cell phone in the center console and completing a brief simulated cell phone conversation, *etc*. The prompted tasks were completed at roughly the same location on the route for each of the participants and were completed only if the participant felt safe in carrying them out.

This video data as well as additional data (such as kinematic data, static and dynamic vehicle segments, trial data, details (sex, skin tone, presence of facial hair, *etc*.) on the participants, frame by frame manually labeled ground truth locations for seven facial landmarks for several trip segments, *etc*.) is what constitutes the full data set. Out of the 48 videos, data for 2 of the static trials and 2 dynamic trials are being withheld (to be possibly released at a future date), bringing the total number of videos (and associated data) in the clipped data set to 44. It is also to be noted that only the data that does not contain personally identifying information has been released publicly. Access to personally identifying data, such as face view videos, is governed by a data sharing agreement. For this reason, any figure in this paper containing a face of a subject who participated in the trials has been masked out using black patches. It is also to be noted that the total number of SHRP2 face view videos (which we work with in this paper) is 41 with 20 videos (and associated data) acquired from static trials and 21 videos (and associated data) acquired from dynamic trials.

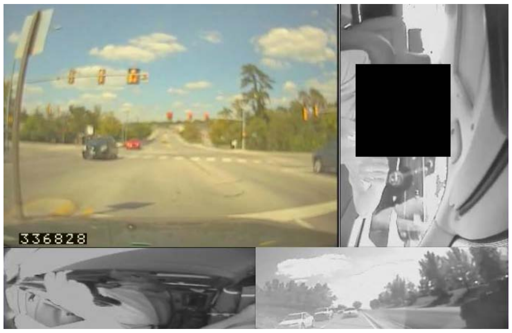


Figure 2: A sample frame showing the standard SHRP2 video views recorded by the SHRP2 configured Data Acquisition System (DAS). This image has been reproduced (with some modifications) from a document providing an overview of the mask head pose validation data that was obtained after signing a data sharing agreement. The face of the subject in the face view portion of the image has been covered with black patches as this information cannot be made public under the terms of the data sharing agreement.

3.2. Details on Our Approach

As is the case with any vision related method, our approach requires a training stage following which the models built can be evaluated on test data. Details on these two stages follow.

3.2.1 Training Stage

In order to build classifier models for the automatic detection of a cell phone in a supervised setting, it is necessary to provide them with consistently labeled training data. Our training data for cases when a cell phone was not in use (negative class data) consisted of frames from video segments where the subject was either seated in a stationary car and performing tasks such as checking the side view mirrors, looking forward, looking at center console, *etc*. or

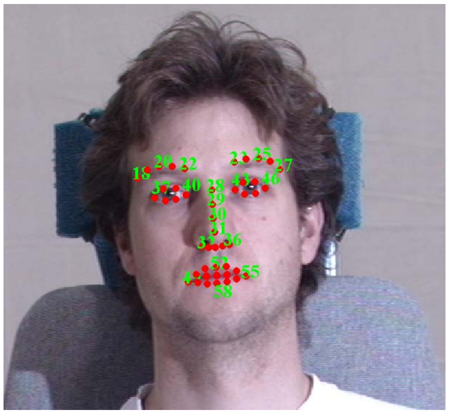
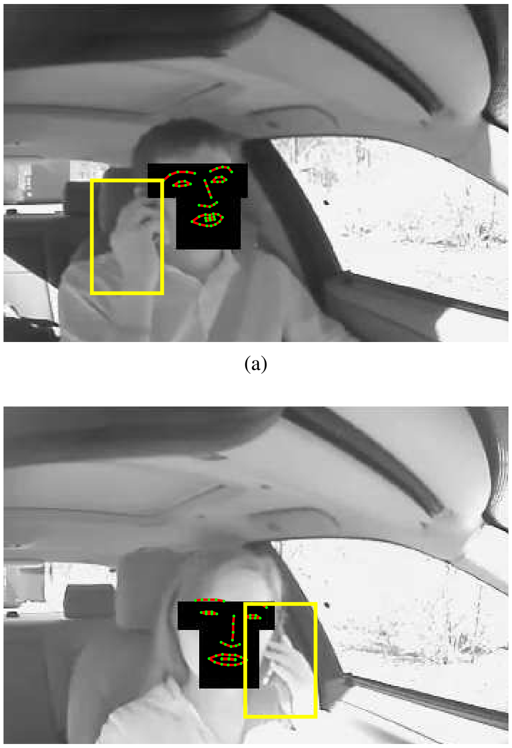


Figure 3: The locations of the 49 facial landmarks that are localized using the SDM algorithm overlaid on an image from the CMU Multi-PIE database [17, 18]. The landmarks were used by us to determine a crop of the region of interest in a frame (image) contained (or did not contain) a cell phone.



(a)

(b)

Figure 4: The process by which crops of the region of interest were generated to check for the presence of a cell phone being held in the (a) right hand of the subject, (b) left hand of the subject. The faces of the subjects have been covered with black patches as this information cannot be made public under the terms of a data sharing agreement.

was driving and performing tasks such as signaling a lane change, checking the speed of the car, turning the radio on or off, looking forward, *etc*. In similar fashion, we also used frames from video segments of the same subjects where the subjects were using a cell phone (with one hand pressed close to one of their ears in order to hold it) in a stationary or moving car. In order to build more accurate models, frames where the subject used their right hand to hold the cell phone were manually separated from those in which in which the subject used their left hand to hold the phone.

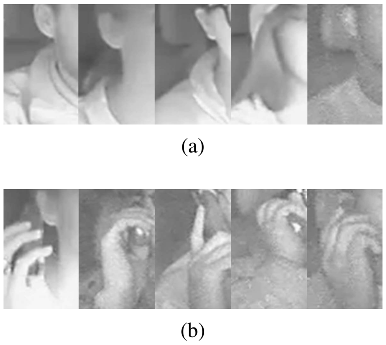
We used an open source implementation [26] of the Supervised Descent Method (SDM) [27] algorithm to extract a crop of the region of interest (containing or not the cell phone) in each of these frames. This implementation of the algorithm is capable of processing video frames at real-time or near real-time speeds on regular laptop or desktop machines and obtain a fairly high level of accuracy. The SDM algorithm was formulated to minimize a Nonlinear Least Squares (NLS) function using descent directions learned from training data and without computing the Jacobian nor the Hessian. For the task of facial alignment, consider an image $\mathbf{d} \in \mathbb{R}^m$ consisting of m pixels with p facial landmarks and with $\mathbf{d}(\mathbf{x}) \in \mathbb{R}^p$ indexing these landmarks. Let $\mathbf{h}$ represent a nonlinear feature extraction technique or function such that $\mathbf{h}(\mathbf{d}(\mathbf{x})) \in \mathbb{R}^{np}$ where n is the dimensionality of the feature vector extracted around each facial landmark (128 dimensional SIFT [20] features are used in this case). If the initial configuration of facial landmarks (generally obtained using a mean shape) can be represented by $\mathbf{x}_0$, then the facial alignment problem is posed as the minimization of the function f over the variable $\Delta\mathbf{x}$, as shown in equation (1).

$$f(\mathbf{x}_0 + \Delta\mathbf{x}) = \|\mathbf{h}(\mathbf{d}(\mathbf{x}_0 + \Delta\mathbf{x})) - \mathbf{\Phi}_*\|_2^2 \qquad (1)$$

In equation (1), $\mathbf{\Phi}_* = \mathbf{h}(\mathbf{d}(\mathbf{x}_*))$ represents the features extracted from a manually labeled training image. $\mathbf{\Phi}_*$ and $\Delta\mathbf{x}$ are known for all training images and hence the goal of SDM is to use this information to learn a series of descent directions to produce a series of updates ($\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$) starting from $\mathbf{x}_0$ and converging to $\mathbf{x}_*$ and then applying these update rules to minimize f when applied to a test image.

The SDM based facial landmark tracking algorithm utilizes the Viola-Jones face detection [25] algorithm in order to detect the subject's face in the first frame of a video and subsequently uses facial landmarks localized in this frame as initialization for the next frame, and so on. The 49 facial landmarks localized by the algorithm are shown in Figure 3. We manually inspected all training data frames in order to ensure that the localization of landmarks was correct.

The next step in our training stage involved the genera-



(a)

(b)

Figure 5: Sample crops of the region of interest generated to train various classifiers for cases when (a) the subject did not have a cell phone in either hand, (b) the subject had a cell phone in his/her right hand.

tion of crops of the region of interest for both the positive and negative class cases using the facial alignment results. We used 50×80 rectangular crops with landmark 18 as the top right corner of the crop region in order to generate positive and negative class crops for cases where subjects were holding (or not holding) a cell phone in their right hand. In similar fashion, 50×80 rectangular crops with landmark 23 as the top left corner of the crop region were generated for cases where subjects were holding (or not holding) a cell phone in their left hand. Use of such crops with reference provided by an interior facial landmark ensured more stability and less variance than crops that would be obtained using a facial landmark along the facial boundary as a reference point as these landmarks are usually localized with higher error and exhibit higher variance even in manually clicked ground truth data [5]. Figure 4 shows how these crops are generated while sample crops generated for cases where a cell phone was not being held and cases when a cell phone was being held in the right hand are shown in Figure 5.

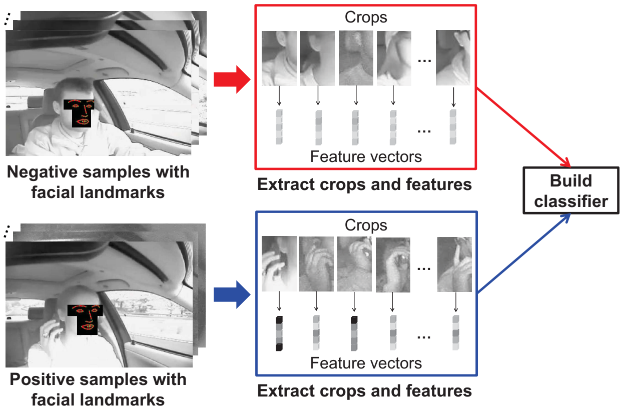

Negative samples with facial landmarks
Crops
...
Feature vectors
Extract crops and features
Build classifier
Positive samples with facial landmarks
Crops
...
Feature vectors
Extract crops and features

Figure 6: The process followed by us to train a classifier that can distinguish between cases when a cell phone is being held close to the right ear of a subject and cases when no cell phone is being held up to the right ear. A similar process was used to train another classifier (using the same corresponding algorithm) that can distinguish between cases when a cell phone was being held close to the left ear of a subject and cases when no cell phone was being held up to the left ear. The faces of the subjects have been covered with black patches as this information cannot be made public under the terms of a data sharing agreement.

```
Algorithm 1: Overview of the Real AdaBoost Algorithm
    Input: Training samples and labels
    $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N))$ where $\mathbf{x}_i \in \mathbb{R}^M$ and
    $y_i \in \{-1, +1\}$ and initial weights for the samples
    $w_i = 1/N \quad i = 1, \ldots, N$
    Output: Ensemble classifier $h(\mathbf{x}) = \text{sign}[\sum_{t=1}^{T} f_t(\mathbf{x})]$
    for $t = 1, \ldots, T$ do
        Fit the classifier to obtain a class probability estimate
        $p_t(\mathbf{x}) = P_w(y = 1|\mathbf{x}) \in [0, 1]$ using weights $w_i$ on
        the training data.
        Set $f_t(\mathbf{x}) \leftarrow \frac{1}{2}\log\frac{p_t(\mathbf{x})}{1-p_t(\mathbf{x})} \in \mathbb{R}$.
        Set $w_i \leftarrow w_i \exp[-y_i f_t(\mathbf{x}_i)] \quad i = 1, \ldots, N$ and
        re-normalize so that $\sum_{i=1}^{N}(w_i) = 1$.
    end for
    Output the ensemble classifier $h(\mathbf{x}) = \text{sign}[\sum_{t=1}^{T} f_t(\mathbf{x})]$.
```

The final stage in the training process was the extraction of features from the positive class (holding a cell phone) and negative class (not holding a cell phone) cases and the building of classifiers using these features. We utilized two different feature representations. When we used raw pixels as features, the feature vectors were 4000 dimensional and were normalized to be unit norm vectors. We also utilized Histogram of Oriented Gradients (HOG) [12] feature descriptors that have been proven to be quite effective in object detection and recognition tasks [14]. We utilized HOG descriptors generated with a spatial bin size of 10 and with 9 orientation bins resulting in a 1008 dimensional feature vector. We benchmark the performance obtained using these two feature descriptors in conjunction with different classifiers, the first of which is the Real AdaBoost [16] framework of ensemble classifiers. We chose the Real AdaBoost classifier due to the minimal parameters that need to be determined to utilize it (only the number of boosting rounds or number of classifiers in the ensemble need to be specified) and its resistance to overfitting [15, 21]. The Real AdaBoost framework not only allows for the classification of a feature

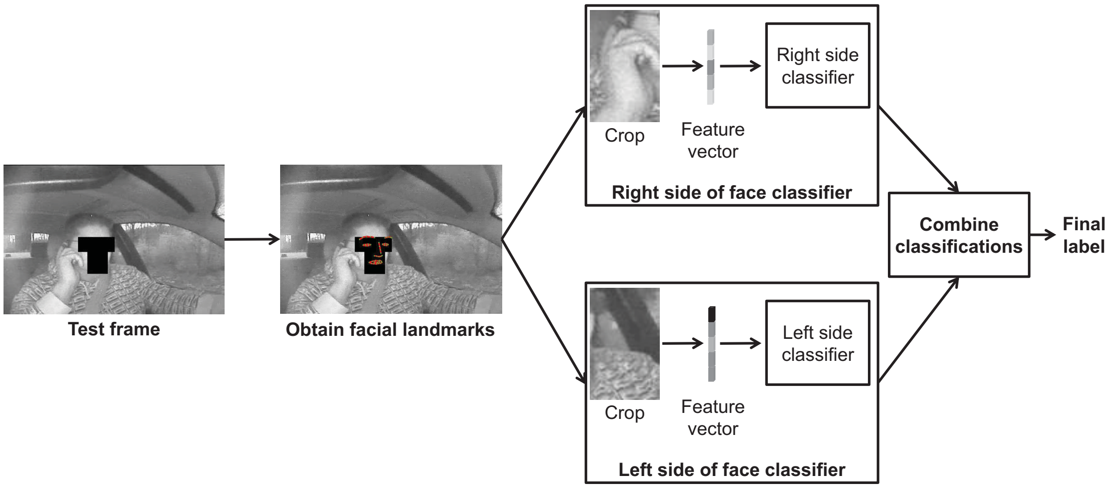


Figure 7: The process followed by us to determine if the subject in a test frame was using a cell phone (holding it close to his/her right/left ear) or not. The faces of the subjects have been covered with black patches as this information cannot be made public under the terms of a data sharing agreement.

vector as positive or negative, but also returns a confidence score for the prediction. This allows us to construct Receiver Operating Characteristic (ROC) curves to summarize performance. Since the exact details of the Real AdaBoost algorithm are slightly different from the more commonly known discrete AdaBoost version, we provide a brief summary of the algorithm.

Let $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N))$ denote a set of N training examples where $\mathbf{x}_i \in \mathbb{R}^M$ is a set of feature vectors and $y_i \in \{-1, +1\}$ is a set of labels for the features vectors in a binary classification problem. Given these training samples, along with a set of weights w_i for each data sample over the indices of S, *i.e.*, over $\{1, \ldots, N\}$, the Real AdaBoost algorithm is an ensemble learning method that aims at combining a set of weak learners or classifiers $f_t(\mathbf{x})$ to form a stronger prediction rule. In the most general form $f_t(\mathbf{x})$ has the form $f_t(\mathbf{x}) : \mathbb{R}^M \rightarrow \mathbb{R}$. Boosting uses the weak learners repeatedly over a set of rounds $t = 1, \ldots, T$ with different weights for the training examples that are updated after each round based on which samples are correctly or incorrectly classified. It is to be noted that the sign of $f_t(\mathbf{x})$ can be interpreted as the predicted label (-1 or $+1$) to be assigned to instance $\mathbf{x}$, and the magnitude of $f_t(\mathbf{x})$ ($|f(\mathbf{x})|$) as the confidence in the prediction. When decision trees are used as the weak learners, this form of Real AdaBoost coincides with one of the forms of the generalized AdaBoost algorithm outlined by Schapire and Singer in [21]. The Real AdaBoost algorithm is outlined in Algorithm 1 which is reproduced from [16] with minor changes to notation.

The other classifiers we use are Support Vector Machines (SVMs) [11] with a Radial Basis Function (RBF) kernel and a random forest [8]. These classifiers can also be configured to return a value that can be interpreted as confidence score of their class prediction (a probability value in the case of SVMs and the number of trees that vote for a class label in the case of the random forest). We built two different sets of classifiers to better deal with the problem of the cell phone being held in different hands. Figure 6 provides an overview of the training process.

3.2.2 Testing Stage

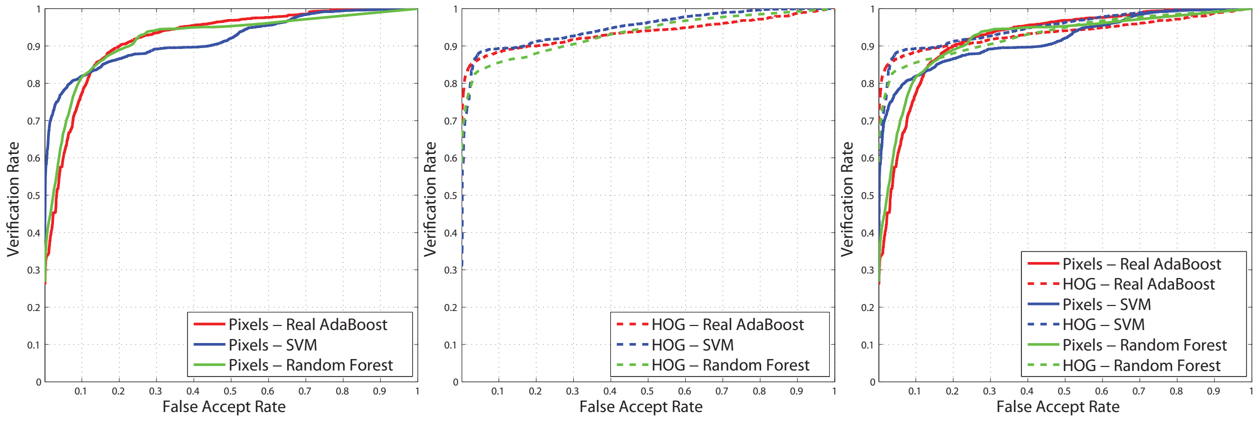
During the test stage of our algorithm, a similar set of steps to those previously described in the training stage were used to extract region of interests in an input frame to determine if a cell phone was present in the extracted regions or not. Again, we utilized the SDM algorithm to localize facial landmarks and generate two crops on the right and left sides of the face in order to check for cell phone presence. Features extracted from these crops were classified using the appropriate (right or left side) side classifiers and the frame was labeled as not having a cell phone present only if both classifiers returned a negative result while in all other cases it was labeled as containing a cell phone. Figure 7 illustrates the sequence of steps followed during the test stage in order to determine if the subject in a test frame is using a cell phone or not.

4. Experiments and Results

Our training data for cases when a cell phone was not in use (negative class data) consisted of 1479 frames obtained

Table 1: A summary of our experimental results. The speed of fitting in Frames Per Second (FPS), the Verification Rates (VRs) at various False Accept Rates (FARs), Equal Error Rates (EER), Area Under the ROC Curve (AUC), and the classification accuracy rates obtained are listed for each feature extraction technique and classification algorithm combination. The best values for each evaluation measure are indicated in bold text.

Approach	FPS	VR @ 0.1% FAR	VR @ 1% FAR	VR @ 10% FAR	EER	AUC	Accuracy
Pixels – Real AdaBoost	**7.5**	0.262	0.414	0.773	0.142	0.920	0.844
HOG – Real AdaBoost	**7.5**	**0.695**	**0.819**	0.883	0.112	0.935	**0.939**
Pixels – SVM	3.7	0.437	0.654	0.819	0.152	0.917	0.787
HOG – SVM	6.0	0.317	0.708	**0.893**	**0.105**	**0.949**	0.842
Pixels – Random Forest	**7.5**	0.298	0.416	0.816	0.148	0.918	0.801
HOG – Random Forest	**7.5**	0.600	0.733	0.857	0.141	0.933	0.927



(a) (b) (c)

Figure 8: ROC curves obtained using three classifiers and (a) raw pixels as features, (b) HOG features, and (c) both raw pixels and HOG features.

from 30 video segments of 11 subjects. We also used 489 frames obtained from 20 video segments of the same 11 subjects where the subjects were using a cell phone. Only one of the subjects (10 video segments and 137 frames) used his/her left hand to hold the cell phone while in all other cases the right hand was used to hold the cell phone. This was reflection of the skew in the data collected as only a few subjects used their left hand to hold a cell phone when requested to do so. This data was used to extract normalized pixel and HOG feature descriptors and build classifier models. All our training and test code was implemented using MATLAB code and mex functions. Our Real AdaBoost ensemble was built using 100 weak decision trees of depth 2 and implemented using an open source toolbox [13]. We used 100 trees in our random forest classifier that was again implemented using open source code [19]. Finally, we used the LIBSVM library to build an SVM classifier [9, 10].

Our test data consisted of 9288 video frames of 30 subjects in which the subjects were driving a car or seated in a stationary one and not using a cell phone and a corresponding set of 3735 frames in which the same subjects were using a cell phone. Thus, the total number of test frames was 13023, making our study more comprehensive than the one carried out in [4]. The subject held a cell phone in his/her left hand in only 429 frames out of the 3735 frames in which a cell phone was being used. It must be noted that this set of frames was retained for evaluation from a larger set of frames from which some frames were discarded due to the fact that no face was detected in them, meaning that a crop of the region of interest could not be generated in these frames as the SDM algorithm could not be used to obtain the coordinates of facial landmarks of the subject in the frames. It must also be noted that there was no overlap of subjects, and hence video frames, between the training and test data used in our study.

Figure 8 shows the ROCs obtained using the various classifiers and feature extraction techniques and Table 1 summarizes the key results obtained as part of our study. As can be seen from the table, HOG features provide for a more robust representation and result in higher classification accuracy rates, Area Under the Curve (AUC) values, and higher Verification Rates (VRs) at various False Accept Rates (FARs) for all three classifiers with the combination of AdaBoost and HOG features resulting in the highest classification rate of 93.86%. Thus, our results are promising and competitive with those obtained in similar studies carried out by Artan *et al.* [4] (highest classification rate of 86.19%) and Zhang *et al.* [29] (highest classification rate of 91.20%), although it must be noted that each study utilized different training and testing data. However, our study is far more thorough than the previously mentioned ones in that our tests are carried out over a much larger set of images and also in the choice of data used for evaluation, which was acquired using strict protocols by a government agency for a specific purpose. While portions of the data (including that used by us in our study) acquired can not be made public under terms of a data sharing agreement, it is our hope that presenting our findings will be of use to the research community and further aid in the development of systems aimed at addressing this problem.

The frame rates in Table 1 obtained by the combination of various feature extraction techniques and classifiers were those obtained when run on a desktop computer with an Intel Xeon E5530 processor with a clock rate of 2.40 GHz running Windows 7 Enterprise. The frame rates for are more than acceptable for post-processing of SHRP2 face view video frames, which was the goal of our work.

5. Concluding Remarks and Future Work

The hazards associated with driver distraction due to cell phone usage have been studied in great detail over the past few years. This has motivated several research efforts aimed at developing algorithms and systems capable of automatically detecting when a cell phone is being used by a driver, *i.e.*, holding it close to one of his/her ears. We have presented a robust framework for a vision based automated cell phone detection system and performed a thorough evaluation of our approach on challenging low resolution SHRP2 face view videos from the head pose validation data that was acquired for of a study of naturalistic driving behavior. Our system utilizes the SDM based facial landmark tracking algorithm to localize a dense set of facial landmarks in frames from a video sequence and then extract a crop of the region of interest which can then be classified as containing a cell phone or not. We evaluated our approach using different combinations of feature extraction techniques and classifiers and obtained promising and accurate results using all these combinations. This is, to the best of our knowledge, also the first such evaluation carried out on this relatively new data.

A limitation of the SDM based facial landmarking algorithm (that can result in the generation of poor crops of the region of interest for cell phone detection which can affect the accuracy rates of all classifiers in our approach) is that it can only track facial landmarks across faces that exhibit a yaw (turning of the head) angle between -45° and $+45^\circ$. It is also sensitive to the initialization provided by the face detector results that in turn can be affected under harsh illumination conditions (excessive sunlight or darkness), the presence of occlusions (hair covering the face, a cap, *etc.*), *etc*. This restricted the scope of our study to video sequences with moderate illumination conditions in which the subject of interest did not exhibit excessive facial pose variation or excessive facial occlusion. One area of future research will involve investigations into how our approach can be improved or modified using alternative combinations of face detection and facial alignment algorithms that can provide highly accurate results even in the presence of larger pose variation, larger facial occlusion levels, and harsher illumination conditions than those the SDM based facial landmarking algorithm can tolerate. Another area of work that we wish to investigate regards improvements that could be made to our system to ensure that it exploits parallelization of tasks and GPUs. Such a system would be of even greater use to researchers in automatically annotating video data sets in order to asses driver behavior, as was the aim in this work, or could also suitably modified and deployed in a real-world scenario to monitor drivers and help in decreasing the number of car crashes due to distracted driving.

Acknowledgements

This work was supported by the Federal Highway Administration's Exploratory Advanced Research Program under contract DTFH61-14-C-00006.

References

[1] Official US Government Website for Distracted Driving. http://www.distraction.gov/stats-research-laws/facts-and-statistics.html.

[2] Strategic Highway Research Program (SHRP2). http://www.trb.org/StrategicHighwayResearchProgram2SHRP2/Blank2.aspx.

[3] Virginia tech transportation institute (VTTI). http://www.vtti.vt.edu/.

[4] Y. Artan, O. Bulan, R. P. Loce, and P. Paul. Driver Cell Phone Usage Detection from HOV/HOT NIR Images. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 225–230, June 2014.

[5] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing Parts of Faces Using a Consensus of Exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 545–552, June 2011.

[6] C. Bo, X. Jian, X. Li, X. Mao, Y. Wang, and F. Li. You're Driving and Texting: Detecting Drivers Using Personal

Smart Phones by Leveraging Inertial Sensors. In *Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 199–202, Sept. 2013.

[7] D. S. Breed and W. E. Duvall. In-Vehicle Driver Cell Phone Detector. http://www.google.com/patents/US8731530, May 2014. US Patent 8731530 B1.

[8] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, Oct. 2001.

[9] C.-C. Chang and C.-J. Lin. LIBSVM – A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[10] C.-C. Chang and C.-J. Lin. LIBSVM – A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27:1–27:27, Apr. 2011.

[11] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, Sept. 1995.

[12] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, June 2005.

[13] P. Dollár. Piotr's Computer Vision Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9):1627–1645, Sept. 2010.

[15] Y. Freund and R. E. Schapire. A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, Sept. 1999.

[16] J. Friedman, T. Hastie, and R. Tibshirani. Additive Logistic Regression: a Statistical View of Boosting. *Annals of Statistics*, 28(2):337–407, Dec. 2000.

[17] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. In *IEEE International Conference on Face and Gesture Recognition (FG)*, pages 1–8, Sept. 2008.

[18] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image and Vision Computing*, 28(5):807–813, May 2010.

[19] A. Jaiantilal. randomforest-matlab. http://code.google.com/p/randomforest-matlab/.

[20] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, Nov. 2004.

[21] R. E. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, 37(3):297–336, Dec. 1999.

[22] D. L. Strayer and F. A. Drews. Cell-Phone–Induced Driver Distraction. *Current Directions in Psychological Science*, 16(3):128–131, 2007.

[23] D. L. Strayer, F. A. Drews, and D. J. Crouch. A Comparison of the Cell Phone Driver and the Drunk Driver. *Human factors: The Journal of the Human Factors and Ergonomics Society*, 48(2):381–391, 2006.

[24] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, June 2001.

[25] P. Viola and M. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, May 2004.

[26] X. Xiong and F. De la Torre. IntraFace (Matlab Functions). http://www.humansensing.cs.cmu.edu/intraface/download_functions_matlab.html.

[27] X. Xiong and F. De la Torre. Supervised Descent Method and its Application to Face Alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 532–539, June 2013.

[28] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin. Detecting Driver Phone Use Leveraging Car Speakers. In *Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 97–108, Sept. 2011.

[29] X. Zhang, N. Zheng, F. Wang, and Y. He. Visual Recognition of Driver Hand-held Cell Phone Use Based on Hidden CRF. In *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 248–251, July 2011.

[30] X. Zhu and D. Ramanan. Face Detection, Pose Estimation, and Landmark Localization in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886, June 2012.